Migrating SMF from Data Set Recording to Log Stream Logging

E. Ozan Baran
z/OS Integration Test

Andrew M. Sica
z/OS System Logger Level 3

This document can be found on the web, www.ibm.com/support/techdocs under the category of “White Papers.”

Beginning with z/OS V1R9, the System Management Facility (SMF) can be configured to use System Logger to write records to log streams. This paper gives an overview of the benefits of migrating SMF to log streams, reviews performance considerations and will describe a migration process using the z/OS Integration Test environment as an example.

Advantages to recording SMF data in Log Streams

The SMF exploitation of System Logger services provides many benefits to the user: [1]

Performance

With SMF support of log streams, SMF data is captured faster than if using MANx data sets. In addition, since System Logger manages data flow and available storage, there is no concern over buffer overrun due to MANx data set switch processing.

This support allows for more efficient dumping, as dump processing can be run against a given log stream which may hold just a subset of your SMF data (as described in the next section).

Manage SMF data on a per log stream basis

SMF allows the user to determine what SMF records are sent to a given log stream on a per system basis. This allows more customization on how SMF records are managed and grouped: users could choose to merge SMF data from multiple systems into a single log stream to give a sysplex view; isolate certain SMF record types to a particular log stream; or group certain types together as best suits their environment.

This ability to filter SMF data on a log stream basis makes dump processing more efficient as well. Dump programs can be run against the log stream holding the SMF types you are interested in - it is not necessary to crawl through unrelated data.

Data reliability

System Logger protects an exploiter's data against a single point of failure. By using a log stream to store real-time data, SMF takes advantage of the data reliability provided – further, this mechanism is managed by Logger processing, and is of no functional impact to the exploiting application.

Further details discussed in "SMF Recording with MVS Logger"; Ahmad. McDonough

To explain this further, an understanding of the flow of data once it is written to a log stream is needed. Here is a brief overview:

Data written to a log stream is kept in interim storage until it is offloaded to DASD log data sets. The actual storage mediums used depend on the type of log stream:

- CF structure log streams store log data in a coupling facility list structure.
- DASD-only log streams store log data in data space local buffers.

While data is in interim storage, Logger manages a duplex copy of the data. For DASD-only log streams, log stream data is duplexed to staging data sets. CF structure based log stream data may be duplexed to a variety of storage mediums (staging data sets, local buffers, or another XES structure via system managed duplexing rebuild). The duplex copy of data will serve as a backup should the primary storage copy be lost.

Once an offload is triggered (caused by a high data threshold being reached, for example), data is written out to more permanent DASD log data sets and scratched from interim storage.

It is important to note the actual location of log data is of no concern to an application attempting to read it – Logger browse (read) processing manages this process, and it is abstract to the exploiter.

This management helps ensure data is recoverable should a system failure or disaster occur. This topic will be touched on again briefly when the test environment is discussed in “Determining Log Stream configuration for the Integration Test environment”.

For detailed information on setting up z/OS log streams, see *z/OS Setting up a Sysplex Chapter 9*

Browsing (Dumping) SMF data

When it is necessary to dump SMF data, the IFASMFDL dump job is used. Simply specify the appropriate log stream name(s) along with the dates and times being requested.

The IFASMFDL program dumps the log stream data to sequential data sets which can be used to produce reports.

Data Retention and Deletion

Data retention is managed by System Logger on an individual log stream basis. This allows the installation to determine how long record types in a particular log stream are kept.

For SMF log stream data, the user controls how long data is managed by specifying the log stream definition parameters RETPD and AUTODELETE:

- RETPD*(days)* – specifies the number of days SMF data should be retained in the log stream. After this period expires, data is eligible for deletion. For example, specifying RETPD=365 will cause data to retained for one year before it can be deleted.

- AUTODELETE*(YES | NO)* – When AUTODELETE=YES is specified, System Logger will automatically delete log data for which the retention period has expired. If AUTODELETE=NO is specified, SMF data will not be deleted automatically when eligible.

Configuration Considerations

Choosing a CF Structure Log Streams or DASD-only Log Streams

There are many factors to consider when deciding whether to use a CF structure log stream or DASD-only log stream. In the “Data Reliability” topic, differences in log stream type interim storage and data flow were discussed. Another important consideration in planning is the scope of the SMF data you wish to record in a given log stream:

- CF structure log streams can be connected and written to by multiple systems concurrently. So if your installation has a requirement to write SMF data from multiple systems into a single log stream you must use a CF log stream.
- If each of your log streams is going to be written to by a single system then you can choose to use either CF or DASD-only log streams.

Note: DASD-only log streams can only be connected to from one system at a time.

There are many other factors to consider when choosing between the two types of log streams and your decision should be based on your environment. If you are not familiar with System Logger and log streams, see *z/OS MVS Setting Up a Sysplex* for more information.

Determining log stream configuration for the test environment:

For the test environment, both CF structure and DASD-only log streams are used. SMF type data was grouped as follows:

DASD-only log streams (One per system):

- SMF Type 0-29 records in one log stream.
- SMF Type 70-79 records in a second log stream.
- SMF Type 30 records only in a third log stream (as the test environment cuts them at a high rate).
- For all other records, a default log stream was created.

CF Structure log stream (Written to by ALL systems):

- SMF Type 88 records from all systems in one log stream

For example, on system SYS1, the following log streams were created:

```
IFASMF.SMF0T029.SYS1
IFASMF.SMF70T79.SYS1
IFASMF.SMF30.SYS1
IFASMF.SMFDFLT.SYS1
IFASMF.SMF88.PLEX2
```

A CF structure called `IFASMF_SMF88` was created for the `IFASMF.SMF88.PLEX2` log stream.

The topic of planning log stream configuration is discussed in detail in various publications – such as the *IBM Redbook "System's Programmers Guide to: z/OS System Logger", or z/OS Setting up a Sysplex Chapter 9.4.2.3 "Determine Which Log Streams Map to Which Coupling Facility Structures".*

Estimating Interim Storage, Offload and Staging Data Set Sizes

Interim Storage for DASD-only log streams

Planning information for interim storage is best described in: *IBM Redbook* "*System's Programmers Guide to: z/OS System Logger*".

"For DASD-only log streams, System Logger uses local buffers in System Logger's data space for interim storage. System Logger will then duplex the data simultaneously to staging data sets. Unlike CF-Structure based log streams, you have no control over this processing; System Logger always uses this configuration for DASD-only log streams."

Interim Storage for CF structure log streams

For IFASMF.SMF88.PLEX2, the CF-based log stream, the interim storage is a CF structure. To calculate the appropriate structure sizes for most System Logger exploiters, use the CFSizer tool available at:

http://www-03.ibm.com/systems/z/cfsizer/

At the time this article was written, z/OS MVS System Management Facilities (SMF) was not yet an exploiter of the CFSIZER tool. Therefore a special method needed to be developed to come up with structure sizes for the installation.

The goal was to not have SMF type 88 data to be sitting in the CF for long. Therefore a small CF structure was needed. To determine a good structure size the rate at which SMF type 88 data was being written in a day was determined and then estimates were made to set what size structure to create.

For example, on this test system 6 cylinders worth of SMF type 88 data were being written a day. Since this is just an estimation an assumption was made that all of the images in the sysplex are writing the same amount of SMF type 88 data.

In this test sysplex there are four systems. SMF type 88 data is going to be written to the IFASMF.SMF88.PLEX2 CF structure log stream from each one of these four systems. Therefore the number of cylinders SMF used for type 88 data was multiplied by 4 to estimate the amount of space the structure IFASMF_SMF88 needs per day:

Structure	Size (cyls)	Size (MB)
IFASMF_SMF88	24	17 MB

This means a 17 MB IFASMF_SMF88 structure should hold approximately a day's worth of SMF type 88 data from all four systems. Since it was decided to not leave the SMF data sitting in the CF all day before being offloaded to DASD a structure smaller than 17MB is going to be defined.

Remember when sizing CF structures not all of the space will be available for the log stream data. Some of it contains control information used by the Coupling Facility Control Code (CFCC). System Logger also uses space to store control information related to a given log stream. In the test environment, it was determined these other uses to be roughly 8 MB.

Since all of this data is going to be offloaded to DASD, there is no advantage to having a large structure size. The goal is to have a structure large enough so Logger is not offloading constantly or encountering frequent full conditions.

It is also important to account for spikes in IXGWRITE activity which could potentially also trigger a full condition. Based on these factors it was decided to size the `IFASMF_SMF88` structure to be 15 MB.

SMF type 88 data showed that the IFASMF.SMF88.PLEX2 log stream was being offloaded 3 to 4 times a day.

For configurations where multiple SMF log streams are defined to the same structure (or you are collecting all SMF data is a single log stream), you would likely want to use a larger structure size. Again, there is no advantage to making the structure very large – because the SMF log streams are being used in a funnel-like manner, all data will be offloaded eventually. The structure size should generally be large enough to accommodate the peak level of write activity likely to occur, including short term spikes, without encountering a structure or entry full condition. In other words during peak times the structure needs to be large enough to ensure an offload is not triggered by the HIGHOFFLOAD threshold value.

As mentioned earlier, there is currently no CFsizer tool support for SMF. However, you may want to look at other Logger exploiter recommendations which have similar usage characteristics as an example – such as IMS. *IBM Redbook "System Programmer's Guide to: z/OS System Logger"* talks about other Logger exploiters in detail.

Staging Data Sets

To ensure maximum recoverability, it was decided to define the CF structure log stream `IFASMF.SMF88.PLEX2`, to always use staging data sets as a duplexing medium. This ensures a hardened, failure independent copy of data in interim storage exists on persistent media – protecting data against multiple failures.

The staging data set size (STG_SIZE) was not specified for `IFASMF.SMF88.PLEX2` so System Logger would use the default value, which is the amount specified in the SIZE parameter of the structure definition in the CFRM policy. Basically a staging data set was allocated as large as the CF structure, namely 15 MB.

This was a sufficient size since SMF type 88 data did not show any staging data set full conditions.

Offload Data Sets

The main goal when setting up the offload data set was to ensure the size is large enough to avoid frequent data set switches during offload processing. A rule of thumb to follow is setting the size large enough to handle multiple offloads and not cause a switch during the day.

The offload data sets were sized at 1000 cylinders each to begin with. Use the *LS_SIZE* parameter, specified in 4k blocks, to tell System Logger what size offload data sets should be used for a log stream. To convert from cylinders to 4K blocks:

```
1 cylinder = 180 4K blocks; so the LS_SIZE was defined
as 180000.
```

```
SMF type 88 data showed that 1000 cylinder offload data
sets were large enough to handle a few days' worth of SMF
data stored in the IFASMF.SMF88.PLEX2 log stream.
```

Structure and Log Stream Definitions

Here is the sample structure and log stream definitions in the CFRM and System Logger policies

```
CFRM Policy:
STRUCTURE NAME(IFASMF_SMF88)
SIZE(15360)
DUPLEX(ALLOWED)
PREFLIST(CFAA,CFAB)

SYSTEM LOGGER POLICY:
DEFINE STRUCTURE NAME(IFASMF_SMF88)
LOGSNUM(1) MAXBUFSIZE(65276)
```

A value of 65276 was picked as the MAXBUFSIZE - System Logger documentation suggests picking this size unless you need it to be bigger. SMF publications recommend a value between 33024 and 65532, but for this test the Logger recommendation was used.

This MAXBUFSIZE value results in an ELEMENT size of 256 rather than 512. SMF type 88 data showed that this size, along with the other configuration decisions made, didn't cause any structure and/or staging data set full events resulting in frequent offloads.

```
SYSTEM LOGGER POLICY
CF structure type log stream example:

DEFINE LOGSTREAM
  NAME(IFASMF.SMF88.PLEX2) LS_SIZE(180000)
  STRUCTNAME(IFASMF_SMF88)
  HIGHOFFLOAD(60) LOWOFFLOAD(35)
  AUTODELETE(YES) RETPD(2)
  LOGGERDUPLEX(UNCOND)
  STG_DUPLEX(YES)
  DUPLEXMODE(UNCOND)
  OFFLOADRECALL(NO)
```

The default HIGHOFFLOAD(60) and LOWOFFLOAD(35) values were used. These log streams will be used to write data primarily, and occasionally retrieve it (dump). This means the log stream offload will begin at 60% full and offload data to the 35% full point. The capacity between the HIGHOFFLOAD point and the 100% full mark acts as a buffer – this allows System Logger to keep accepting new write requests while an offload is in progress. Depending on usage characteristics, you may choose to use different values or increase the structure space available. It is important to look at performance related data (discussed later in this paper) and attempt to avoid structure full type conditions. This is important because if available structure space dries up (100% full), System Logger will stop accepting new writes from applications until space can be made available via offload.

As mentioned previously, it was decided to always duplex SMF data to staging data sets to ensure maximum recoverability of log data, so DUPLEXMODE(UNCOND) and STG_DUPLEX(YES) were set.

As mentioned earlier, for other SMF data types only DASD-only log streams are being used. Here is an example of a log stream definition used:

```
SYSTEM LOGGER POLICY
DASD-based type log stream example:

DATA TYPE(LOGR)
DEFINE LOGSTREAM NAME(IFASMF.SMF30.SYS1)
DASDONLY(YES)
STG_SIZE(12800)
LS_SIZE(180000)
AUTODELETE(YES)
RETPD(2)
HIGHOFFLOAD(60)
LOWOFFLOAD(35)
```

Notice a staging data set size of 50 MB (12800 4K blocks) was picked. Similar to the CF structure size, this was based on analysis of the volume of data which was being written and the data's write characteristics.

Based on the requirements for this particular log stream, it was decided SMF data should be retained for 2 days. Thus, the log streams were defined to use AUTODELETE(YES) and RETPD(2).

SMFPRMxx member definition

To activate SMF upon IPL, the following updates to the *SMFPRMxx* parmlib member were made.

```
RECORDING(LOGSTREAM),
DEFAULTLSNAME(IFASMF.SMFDFLT.&SYSNAME),
LSNAME(IFASMF.SMF0TO29.&SYSNAME,TYPE(0:29)),
```

```
LSNAME(IFASMF.SMF30.&SYSNAME,TYPE(30)),
LSNAME(IFASMF.SMF70T79.&SYSNAME,TYPE(70:79)),
LSNAME(IFASMF.SMF88.PLEX2,TYPE(88)),
PROMPT(LIST),

/* Prompt parameter allows you to dynamically switch */
/* between logging and data set recording via SETSMF */
/* command */
```

The *DSNAME* statements for the MANx data sets were left in the parmlib member. By doing this, the MANx data sets are still available if needed to dynamically switch back to SMF data set recording.

```
DSNAME(SYS1.SMF.&SYSNAME..MANS,
       SYS1.SMF.&SYSNAME..MANT,
       SYS1.SMF.&SYSNAME..MANU,
       SYS1.SMF.&SYSNAME..MANV),
```

Migrating a Current Environment to SMF Log Stream Logging

When deciding to exploit SMF log stream logging it is best to migrate systems one or a few at a time thereby providing less disruption to SMF data end users. This migration started with a single system and once successful the rest of the systems were switched to use log streams as well.

Prior to SMF log stream logging, whenever one of the MANx data sets became full, the SMF data would be dumped into a new Generation Data Groups (GDG) data set. These data sets were named similar to:

SMFDATA.SMFSYS1.G1503V00

where SYS1 is the system name and G1503V00 represents the generation and version numbers.

The goal was for the migration was to not change the location of the SMF data and the data set names. This would allow end users to run their jobs to post process the SMF data without changing them or via very minor changes.

There are a few different ways to accomplish this, the following two were considered:

1) Dump when needed
This was probably the easiest method and the one which is recommend but it would have required the end users to run an additional job and possibly to make some minor changes to their jobs.

Basically whenever the end users wanted to look at SMF data, they would have to dump the data they are looking for from the log streams into a data set using the same naming convention their jobs take as input. See the "SWITCH SMF & Run dump program" section below for an example job which does this function.

2) Dump once a day
A job can be scheduled to run once a day and dump the SMF data from the log streams into a data set. The same naming convention can be followed for the end users' jobs take as input when naming this data set. Then the end users will not have to change any post processing jobs.

For instance, the IBM Tivoli NetView for z/OS automation facilities could be used to submit a job similar to the one below at 1 am every morning on each system. This job runs a REXX program to figure out the previous day's date and creates the control cards for the SMF dump program. Then it executes the IFASMFDL program and dumps the SMF data out to a GDG data set.

Here is the sample REXX program:

```
/*** REXX ***/
SysID = MVSVAR('SYSNAME')

jday = DATE('Base')
jdayyest = jday - 1
jyear = LEFT(DATE('Standard',jdayyest,'Base'),4)
jdate = jyear || RIGHT(DATE('Days',jdayyest,'Base'),3,'0')

Queue "         LSNAME(IFASMF.SMF70T79."||SysID||",OPTIONS(DUMP))"
Queue "         LSNAME(IFASMF.SMF0TO29."||SysID||",OPTIONS(DUMP))"
Queue "         LSNAME(IFASMF.SMF30."||SysID||",OPTIONS(DUMP))"
Queue "         LSNAME(IFASMF.SMFDFLT."||SysID||",OPTIONS(DUMP))"
Queue "         OUTDD(DUMPOUT,TYPE(0:255))"
Queue "         ABEND(NORETRY)"
Queue "         DATE("||jdate||","||jdate||")"
Queue "         START("0000")"
Queue "         END("2400")"

"PIPE STACK | PAD 80 | CHOP 80 | > DDNAME=SMFCNTL"
```

Here is the sample JCL:

Below, GETDATE is the above REXX exec and OZ2.REXX is the library it is located in.

```
//OZST JOB 'OZAN',MSGCLASS=A,CLASS=A
//*****************************************************
//* BUILD THE CONTROL CARDS FOR THE SMF DUMP PGM
//*****************************************************
//GETDATE  EXEC PGM=IKJEFT01,
//         DYNAMNBR=50,
//         PARM='%GETDATE'
//SYSPROC   DD  DISP=SHR,DSN=OZ2.REXX
//SMFCNTL   DD  DISP=(NEW,PASS,DELETE),DSN=&&SMFCTL,
```

```
//           SPACE=(TRK,(1,1)),UNIT=SYSDA,VOL=SER=,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040)
//SYSIN      DD  DISP=(NEW,PASS,DELETE),
//           SPACE=(TRK,(1,1)),UNIT=SYSDA,VOL=SER=,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040)
//SYSTSIN    DD  DUMMY
//DUMPOUT    DD  DUMMY
//SYSTSPRT   DD  SYSOUT=*
//*******************************************************
//* ALLOCATE THE NEXT GDG ENTRY
//*******************************************************
//ALLOC1    EXEC PGM=IEFBR14,COND=(4,LT)
//DUMPOUT    DD DSN=OZ2.TEMP(+1),
//           DISP=(NEW,CATLG,DELETE),
//           DCB=(SMFDATA.MODEL.DSCB),
//           UNIT=LOGS,
//           SPACE=(CYL,(750,750))
//*******************************************************
//* DUMP THE SMF DATA
//*******************************************************
//DUMP1     EXEC PGM=IFASMFDL,COND=(4,LT)
//DUMPOUT    DD DSN=OZ2.TEMP(+1),DISP=OLD,
//              SPACE=(CYL,(750,750),RLSE),
//              DCB=(SMFDATA.MODEL.DSCB)
//SYSIN      DD DISP=(OLD,DELETE),DSN=&&SMFCTL
//SYSPRINT   DD SYSOUT=*
//*******************************************************
//* NOTE THE NAME OF THE NEWEST GDG FOR FUTURE REFERENCE
//*******************************************************
//GDGLIST EXEC PGM=SMFGDG,COND=(4,LT)
//STEPLIB    DD DSN=USER.LINKLIB,DISP=SHR,
//       VOL=SER=CMNSTC,UNIT=3390
//SYSUDUMP   DD SYSOUT=*
//SMFGDG     DD DSN=OZ2.TEMP(+1),DISP=SHR
//LOG        DD DSN=OZ2.GDG.LIST,DISP=SHR
```

Switching from SMF Data Set Recording to SMF Log Stream Logging

Once you have the SMFPRMxx *par*mlib member ready, you can switch in a few different ways.

1. IPL with the SMF parmlib member updated with log stream information shown above.
2. Issue the "SET SMF=xx" command and specify the SMFPRMxx parmlib member to switch dynamically.
3. Run the "SETSMF RECORDING(LOGSTREAM)" command to switch dynamically.

If the PROMPT parameter in the SMFPRMxx member is specified as ALL or LIST, regardless of which method is chosen, a WTOR is issued. Respond with U to keep the options in the parmlib member you specified.

The first time through, a dynamic switch was done using the second option. The command "SET SMF=SYS1" was issued and the following messages were received:

```
IEE967I 07.12.02 SMF PARAMETERS 849
MEMBER = SMFPRMSYS1
MULCFUNC -- DEFAULT
LISTDSN -- DEFAULT
STATUS(010000) -- DEFAULT
MAXDORM(3000) -- DEFAULT
DDCONS(YES) -- DEFAULT
LASTDS(MSG) -- DEFAULT
NOBUFFS(MSG) -- DEFAULT
INTVAL(30) -- DEFAULT
DUMPABND(RETRY) -- DEFAULT
REC(PERM) -- DEFAULT
ACTIVE -- DEFAULT
BUFSIZMAX(0256M) -- PARMLIB
BUFUSEWARN(80) -- PARMLIB
SYNCVAL(00) -- PARMLIB
SYS(EXITS(IEFUSI)) -- PARMLIB
SYS(EXITS(IEFUJV)) -- PARMLIB
SYS(EXITS(IEFU85)) -- PARMLIB
SYS(EXITS(IEFU84)) -- PARMLIB
SYS(EXITS(IEFU83)) -- PARMLIB
SYS(EXITS(IEFU29)) -- PARMLIB
SYS(EXITS(IEFUJI)) -- PARMLIB
SYS(EXITS(IEFACTRT)) -- PARMLIB
SYS(INTERVAL(SMF,SYNC)) -- PARMLIB
SYS(DETAIL) -- PARMLIB
SYS(TYPE(0,2,3,6:10,14,15,22:24,26,30,32,33,41,42,
  47:48,59,61:69,70:79,80:83,85,88,89,90:91,94,98,
  100:103,108,110, 115:117,120,130,134,148:151,161,
  200,244,245)) -- PARMLIB
SID(SYS1) -- DEFAULT
JWT(2400) -- PARMLIB
MEMLIMIT(00512M) -- PARMLIB
DSNAME(SYS1.SMF.SYS1.MANV) -- PARMLIB
DSNAME(SYS1.SMF.SYS1.MANU) -- PARMLIB
DSNAME(SYS1.SMF.SYS1.MANT) -- PARMLIB
DSNAME(SYS1.SMF.SYS1.MANS) -- PARMLIB
PROMPT(LIST) -- PARMLIB
LSNAME(IFASMF.SMF70T79.SYS1,TYPE(70:79)) -- PARMLIB
LSNAME(IFASMF.SMF30.SYS1,TYPE(30)) -- PARMLIB
LSNAME(IFASMF.SMF0TO29.SYS1,TYPE(0:29)) -- PARMLIB
LSNAME(IFASMF.SMF88.PLEX2,TYPE(88)) -- PARMLIB
DEFAULTLSNAME(IFASMF.SMFDFLT.SYS1) -- PARMLIB
RECORDING(LOGSTREAM) -- PARMLIB
```

The following WTOR message was received and the operator replied U:

```
*7187 IEE357A REPLY WITH SMF VALUES OR U
```

By issuing a "D SMF" command it can be verified SMF is in fact using the log streams.

```
IFA714I 10.53.42 SMF STATUS 604
LOGSTREAM NAME                BUFFERS        STATUS
A-IFASMF.SMFDFLT.SYS1           15069          CONNECTED
A-IFASMF.SMF0TO29.SYS1           7076          CONNECTED
A-IFASMF.SMF30.SYS1              9935          CONNECTED
A-IFASMF.SMF70T79.SYS1          56084          CONNECTED
A-IFASMF.SMF88.PLEX2              0          CONNECTED
```

SWITCH SMF & Run dump program

SMF is also providing a new dump program, called IFASMFDL, to use with your log streams. It can take multiple log streams as input and can format output to multiple data sets. For details on the program please see the *z/OS MVS System Management Facilities (SMF)* manual. Here is sample JCL to execute the program for collecting SMF data.

```
//IFASMFDL JOB
MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A,REGION=0M,
//  NOTIFY=&SYSUID
//DUMP1  EXEC  PGM=IFASMFDL
//OUT1   DD    DSN=OZ.SMF88.SYS1,DISP=(NEW,CATLG,DELETE),
//  SPACE=(CYL,(100,100),RLSE),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
       LSNAME(IFASMF.SMF30.SYS1)
       LSNAME(IFASMF.SMF70T79.SYS1)
       LSNAME(IFASMF.SMF0TO29.SYS1)
       LSNAME(IFASMF.SMFDFLT.SYS1)
       OUTDD(OUT1,TYPE(0:255)),START(0000),END(2400)
//*
```

Overall, the routine for looking at SMF data is still the same:

- Issue the “SWITCH SMF” command to transfer the log stream data from the buffers into the appropriate log streams.
- Run the IFASMFDL dump program to dump the SMF data

Once using the system logger to manage the offloading and archiving of the SMF data installation exit IEFU29L is not being used. If needed the combination of IEFU29L exit and “SWITCH SMF” command can be used to handle the archiving of your SMF log stream data.

Monitoring your configuration

There are samples of these two commands above:

D SMF - will show the log streams SMF is using, the buffer sizes and if SMF is connected to the log streams or not.

D SMF,O - will show the options SMF is currently using just like the “SET SMF=xx” output seen above.

To see the structures your log streams are connected to, their status, and the number of connections issue the command:

```
D LOGGER,L,LSN=IFASMF.SMF30.SYS1

IXG601I   10.55.46  LOGGER DISPLAY 631
INVENTORY INFORMATION BY LOGSTREAM
LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
IFASMF.SMF30.SYS1          *DASDONLY*       000001 IN USE
  SYSNAME: SYS1
    DUPLEXING: STAGING DATA SET
  GROUP: PRODUCTION

NUMBER OF LOGSTREAMS:  000001

D LOGGER,L,LSN=IFASMF.SMF88.PLEX2

IXG601I   10.56.19  LOGGER DISPLAY 676
INVENTORY INFORMATION BY LOGSTREAM

LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
IFASMF.SMF88.PLEX2         IFASMF_SMF88     000001 IN USE
  SYSNAME: SYS1
    DUPLEXING: STAGING DATA SET
  GROUP: PRODUCTION

NUMBER OF LOGSTREAMS:  000001
```

It is possible to display the staging data set your log stream is using, its location, size, and other information via the command:

```
D LOGGER,C,LSN=IFASMF.SMF30.SYS1,D

IXG601I   10.56.57  LOGGER DISPLAY 685
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SYS1
LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
IFASMF.SMF30.SYS1          *DASDONLY*       000001 IN USE
  DUPLEXING: STAGING DATA SET
    STGDSN: IXGLOGR.IFASMF.SMF30.SYS1.PETPLEX2
            VOLUME=P2LG06  SIZE=012960 (IN 4K)    % IN-
USE=003
  GROUP: PRODUCTION
  JOBNAME: SMF       ASID: 001B
    R/W CONN: 000000 / 000001
    RES MGR./CONNECTED: *NONE*    / NO
```

```
    IMPORT CONNECT: NO

NUMBER OF LOGSTREAMS:  000001
```

D LOGGER,C,LSN=IFASMF.SMF88.PLEX2,D

```
IXG601I   10.57.17  LOGGER DISPLAY 697
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SYS1
LOGSTREAM                  STRUCTURE        #CONN  STATUS
---------                  ---------        ------ ------
IFASMF.SMF88.PLEX2         IFASMF_SMF88     000001 IN USE
  DUPLEXING: STAGING DATA SET
    STGDSN: IXGLOGR.IFASMF.SMF88.PLEX2.SYS1
            VOLUME=P2LG03  SIZE=003960 (IN 4K)    % IN-
USE=001
  GROUP: PRODUCTION
  JOBNAME: SMF       ASID: 001B
    R/W CONN: 000000 / 000001
    RES MGR./CONNECTED: *NONE*    / NO
    IMPORT CONNECT: NO

NUMBER OF LOGSTREAMS:  000001
```

Also, when using CF type log streams, these commands can be run to collect structure related information.

D LOGGER,STR,STRN=IFASMF_SMF88

```
IXG601I   10.57.43  LOGGER DISPLAY 704
INVENTORY INFORMATION BY STRUCTURE
STRUCTURE           GROUP              CONNECTED
---------           -----              ---------
IFASMF_SMF88        PRODUCTION
  IFASMF.SMF88.PLEX2                     YES

NUMBER OF STRUCTURES:  000001
```

D XCF,STR,STRNM=IFASMF_SMF88

```
IXC360I  10.58.05  DISPLAY XCF 719
STRNAME: IFASMF_SMF88
 STATUS: ALLOCATED
 EVENT MANAGEMENT: POLICY-BASED
 TYPE: LIST
 POLICY INFORMATION:
  POLICY SIZE    : 15360 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: N/A
  DUPLEX         : ALLOWED
```

```
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF21     CF22
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 04/17/2008 10:53:08
  CFNAME         : CFAB
  COUPLING FACILITY: XXXXXXX.IBM.02.0000000699FF
                     PARTITION: 13   CPCID: 00
  ACTUAL SIZE    : 15360 K
  STORAGE INCREMENT SIZE: 512 K
  USAGE INFO      TOTAL     CHANGED    %
   ENTRIES:         305           5    1
   ELEMENTS:      25038          32    0
  PHYSICAL VERSION: C2423136 D57CB61F
  LOGICAL  VERSION: C2423136 D57CB61F
  SYSTEM-MANAGED PROCESS LEVEL: 8
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 1
```

Another way to monitor the CF structure log streams is by post processing SMF type 88 data. Once SMF logging has been active for a while, perhaps 24 hours, dump the data and create a System Logger Activity Report using the IXGRPT1 macro. Here is a sample IXGRPT1 report for IFASMF.SMF88.PLEX1 log stream (reformatted to fit this paper):

```
   SYSTEM LOGGER ACTIVITY REPORT (IXGRPT1)

   BYT WRITTN  BYT WRITTN  BYT WRITTN
   BY USERS    TO INTERIM  TO DASD     # WRITES
   IXGWRITES   STORAGE                 INVOKED

   BYT DELETD  # DELETES   BYT DELETD  # DELETS
   INTERIM ST  W/O DASD    INTERIM ST      W/
   W/O DASD    WRITE       W/DASD        WRITE
   ____________________________________________
   171595      172288           0         7
```

```
                                          AVERAGE
---# WRITES COMPLETED------               BUFFER
TYPE1     TYPE2     TYPE3                 SIZE

---------------EVENT---------------
OFF-  DASD  STRC NTRY  STG   STG          RE-
LOAD  SHFT  FULL FULL  THLD  FULL         BLD
  7           0          0               24513

  0     0     0    0     0     0           0
```

For more information on IXGRPT1 macro and the report it generates, please see the *z/OS MVS System Management Facilities (SMF)* manual. Information on using this data can be found in the *IBM Redbook* "*System Programmers Guide to z/OS System Logger*".

Since the structure sizes were estimated special attention was paid to the sizing decisions. To monitor the structures the following data in the IXGRPT1 was reviewed.

- # Type 2 and 3 writes: These, at least the Type 3 column, should be 0. If not adjust the structure size. Type 3 writes indicate write requests processed after a structure full has been encountered.

- # offload events: If too frequent, adjust the high and low offload thresholds, or structure size. Note however offloads are not an indicator of a problem. More interesting is the reason the offload is triggered, so you should consider the values of other SMF data – such as the number of structure full events.

- # structure full events: This should be a rare occurrence as well. If this field is frequently greater than 0 consider adjusting the structure size and checking log stream performance data.

- # DASD shifts: These occur every time the system creates an offload data set. This should be a small percentage of the offload events. Otherwise, the offload data sets might be too small.

- # staging threshold was reached: If too frequent, the staging data set might be too small.

- # staging data set full: As with the staging threshold, check the log streams staging data set size, it may be too small. If unsure what it should be, size it similar to the CF structure the log stream is connected to.

References

- IBM Redbook “System Programmer's Guide to: z/OS System Logger”
- IBM White Paper “SMF Recording with MVS Logger” by Riaz Ahmad and Jeff McDonough
- IBM z/OS Manuals:
  - z/OS MVS Setting Up a Sysplex
  - z/OS MVS System Management Facilities (SMF)
  - z/OS MVS System Commands

IBM Corporation
New Orchard Rd.
Armonk, NY 10504
U.S.A

Produced in the United States of America
03/08

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.