Get Your Data Faster …and More Easily – A User Success Story

MR130SN
Mainframe and Application Development

SBBworks

we can

caWorld'10

abstract

At the last CA World, we announced how the integration between CA MICS® and CA SMF Director automates daily processing feeds. This year, learn how a customer has implemented the solution and improved the overall availability of resource management data while at the same time reducing system overhead. This session will take you step by step through the implementation process and describe the benefits that were achieved.

biography

- **Scott Barry,**
  Principal Consultant, SBB*Works*, Inc.
- Twenty-six years experience with the CA MICS and SAS systems, and related information technology resource management disciplines, including chargeback, capacity management, systems performance and enterprise reporting.

agenda

- Introduction
- IBM System z and SMF data-capture challenges
- CA SMF Director – SMF data management strategy
- Benefits for the enterprise with CA SMF Director
- Implementation planning and deployment considerations
- Using CA SMF Director for SMF data extract requests
- Summary and questions

introduction

- SMF data – key reference for system z resource usage
- Challenges continue to exist with managing SMF data
- IBM's Logstream has provided SMF processing relief
- CA SMF Director introduced to manage the SMF process effectively and efficiently

IBM system z and SMF data-capture challenges

- CP processors and LPARs are increasing in size and speed
- SMF data-volume ever-increasing – new technologies, business analysis and retention requirements
- Legacy SMF MANx dump process setup long-ago, unchanged
- No defined technical standard for SMF data management
- Constant need for information access, including SMF data
- Past practice of giving users their own copy no longer feasible

IBM system z and SMF data-capture challenges – "I want some of that SMF data – why, you ask?"

- z/OS systems' data-capture repository – compared to GTF, other proprietary logging structures, LOGREC, external data-feeds
- Systems, application performance, capacity management, audit, workload projection, IT chargeback/allocation
- Special studies: acquisition, industry comparison, IT/regulatory compliance, business-practice due-diligence
- SMF repository now scaling with SMF Log Stream introduction - New SMF management paradigm

IBM system z and SMF data-capture challenges – "I want some of that SMF data – why, you ask?"

- RMF – near real-time workstation data access
- CICS, DB2, IDMS – transaction, subsystem TRACE event data
- MQSeries, TCP/IP, WebSphere, NetView – enterprise network
- Security event logging – SMF type 83 from zLinux for SOA audit
- CA MICS Resource Management, accounting/chargeback
- MXG, Tivoli – SMF-fed reporting repositories
- ISV application logging – SMF user-type range: 128-255

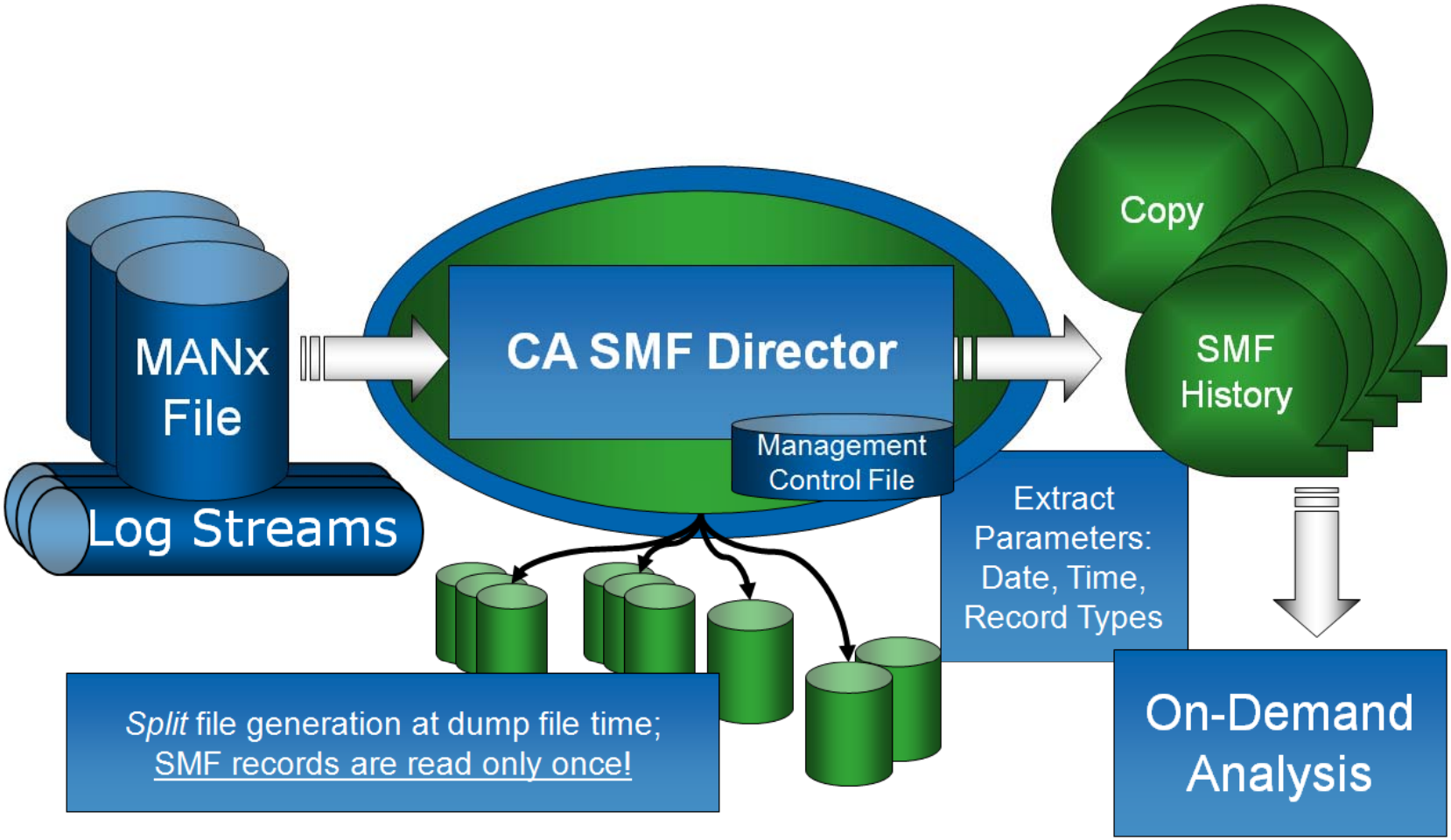
CA SMF Director – SMF management strategy

- Replace legacy SMF MAN file archival process with CA SMF Director
- Enhance SMF data access with SPLIT file feature and SMF-sourced applications, users - CA MICS Incremental Update facility
- Exploit DASD resources for primary SMF logging, not tape
- Reduce redundant SMF data-capture with SMFD EXTRACT requests (DB2 TRACE, studies)
- Limit/control SMF data access using SMFD program, security rules
- Simplify SMF processing control, with minimal operational JCL minimized (dynamic allocation used), and compiled/generated parameter definitions

benefits realized with CA SMF Director

- CA SMF Director's HISTORY file logging can eliminate redundant SMF data capture / duplication (WEEKLY, MONTHLY copies)
- Enhanced operational controls with SMF Logstream migration
- CA SMF Director SPLIT file feature for CA MICS Incremental Update
- Simplified SMF data management administration
- SMF processing cost reduced – storage, CPU, administration

CA SMF Director implementation – system overview



CA SMF Director implementation – planning

- CA SMF Director SCDS – control dataset for SMF dump tracking
- Develop an LPAR migration plan, starting with QA/test LPAR
- Assess alternate history file capture (offsite retention)
- SPLIT file candidates needed? consider EXTRACT requests as alternative, depending on data-volume, retention limit
- Develop SMFD back-out contingency, just in case needed
- Adjust time-based SMF SWITCH/DUMP for SPLIT file capture

CA SMF Director implementation – SCDS configuration, DUMPOPTIONS, SID definitions

- Consider a shared SCDS (SMFD control dataset) per SYSPLEX or shared DASD/tape environment, as required
- SMFD control statements for SCDS and an LPAR (SID) setup:

```
OPTIONS PDEVN(SYSDA) AUTODEL DYNAM OVERLAP(5).
BEGIN CONFIG(01) SID(LPRA) NAME('LPAR PROD A').
DUMPOPTIONS PREFIX(<hlq>.SMFDALL.HIST.LPRA)
  CATLG RETPD(1830).
END CONFIG(01).
```

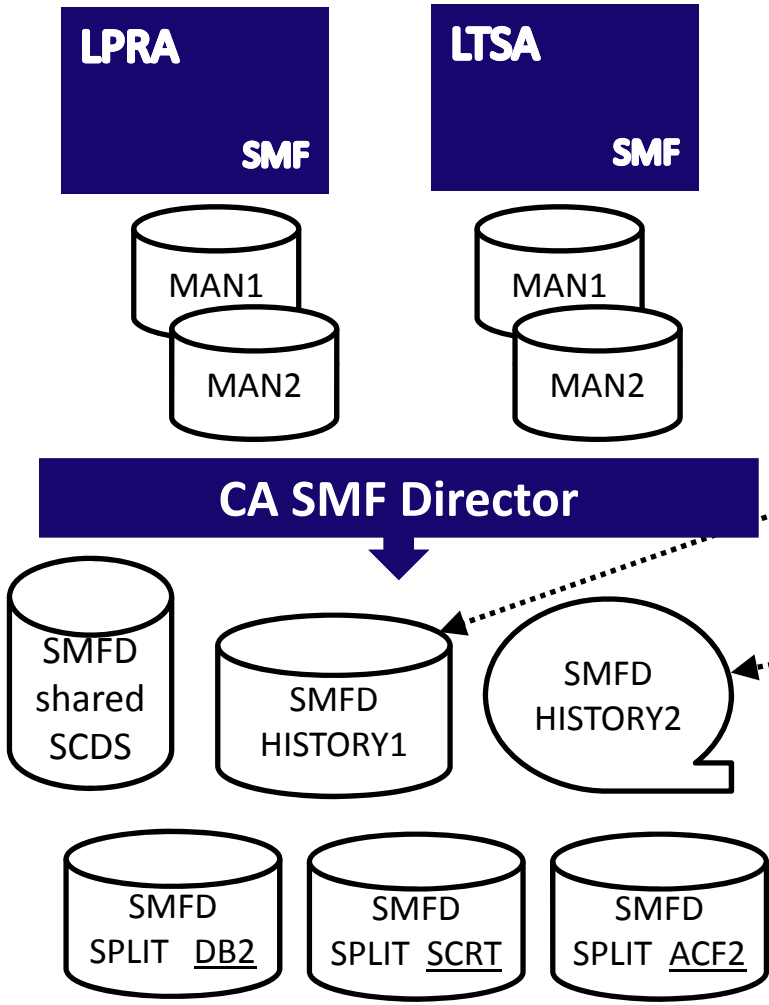
- OPTIONS parameters used with all SID definitions
- DUMPOPTIONS parameters manage each SID – RETPD value = 5+ years
- Each SID definition has its own BEGIN/END group.

CA SMF Director implementation – SMFD DUMP JCL

- SMFD HISTORY and SPLIT files created using dynamic allocation
- SCDS provides allocation attributes in addition to //HISTORYn DDs
- Program SMFD manages SCDS access for multiple, concurrent requests – sample JCL below:

```
//STEP1    EXEC PGM=SMFD,PARM=DUMP
//STEPLIB  DD DISP=SHR,DSN=<hlq>.APFLIB
//SCDS  DD DISP=SHR,DSN=<hlq>.SMFD.PROD.SCDS
//HISTORY1 DD UNIT=SYSDA,SPACE=(CYL,(800,800))
//HISTORY2 DD UNIT=TAPE,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SYSIN    DD DUMMY
```


CA SMF Director implementation – SMFD HISTORY files



SMF MAN dataset archival example:

- Shared SCDS for production LPARs
- One history file created per LPAR, dump
- 'f' identifies the MANx file suffix
- 'nn' sequence count suffix, at MAN file EOV
- New SPLIT file generation per day

• HISTORY primary files – one per SMF DUMP:

```
<hlq>.SMFDALL.SLPRA.Mfmmddyy.ThhmmPnn
<hlq>.SMFDALL.SLTSA.Mfmmddyy.ThhmmPnn
```

• HISTORY archive files (offsite retention, GDPS):

```
<hlq>.SMFDALL.SLPRA.Mfmmddyy.ThhmmAnn
<hlq>.SMFDALL.SLTSA.Mfmmddyy.ThhmmAnn
```

• daily SPLIT files (focus-area content):

```
<hlq>.SMFDDB2.Sxxxx.DAILY(+1)
<hlq>.SMFDRMF.Sxxxx.DAILY(+1)
<hlq>.SMFDACF2.Sxxxx.DAILY(+1)
```


CA SMF Director implementation – EXTRACT request DSN-related SMF types for DSN prefix “PRODAA”

– SMF data request using EXTRACT – select SMF record types related to dataset activity, dataset starts with “PRODAA” :

```
//STEP1    EXEC PGM=SMFD,PARM=READ
//STEPLIB  DD DISP=SHR,DSN=<hlq>.APFLIB
//SCDS     DD DISP=SHR,DSN=<hlq>.SMFD.PROD.SCDS
//EXTRACT  DD DSN=&SYSUID..SMFD.DSNINFO.EXTRACT,
//  DISP=(,CATLG),UNIT=...
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SYSIN    DD *
 EXTRACT SID(LPRA) LWEEK SELECT(14 15 17 18 64)
   WHEN(DSN,EQ,PRODAA*) .
/*
```


CA SMF Director implementation – more EXTRACT request s

– EXTRACT to select SMF record types related to CICS/CMF 110 transactions, VTAM APPLID starts with “CICP”:

```
EXTRACT ALL DAY(mmddyy) SELECT(110:001)
  TODD(EXTRACT) WHEN(CICS,EQ,CICP*) .
```

– EXTRACT to select SMF record type 6 (JES output device) for SMF6UCS value “VPS” (printer study – hex offset 50 + RDW):

```
EXTRACT ALL LMONTH SELECT(6) TODD(EXTRACT)
  WHEN(84,EQ,'VPS') .
```

– EXTRACT to select SMF record type 42 (dataset activity), datasets starting with “SYSP” (hex 94 + 4 byte RDW):

```
EXTRACT ALL YESTERDAY SELECT(42)
  TODD(EXTRACT) WHEN(152,EQ,'SYSP') .
```

CA SMF Director implementation – SPLIT file considerations

- GDG-based SPLIT files use TIMEGDG(…) and TIME(…) parameters for time-interval split processing
- SMFD closes the current generation and creates a new one upon encountering SMF data in the next TIME interval
- The prior-generation TIMEGDG SPLIT file ready for processing by CA MICS or another application when marked as *complete*
- INDEXDSN /INDEXDUP files tracks SMFD-generated split files
- *SMF data for new-interval required to mark current generation as complete – may require SMF SWITCH/DUMP time change*

CA SMF Director implementation – SPLIT file considerations (continued)

- Possible contention when a new generation is created and an existing SPLIT file is allocated, due to SPLIT file allocation using relative generation <hlq>.split_file.SIDxxxx(+1)
- Typical SPLIT file application provides SMF record type/subtype data to users, such as DB2 SMF 102 TRACE, SMF type 80 security data, SCRT monthly reporting, special studies
- Alternative: consider EXTRACT requests rather than SPLIT file definition for current SMF data-retrieval requests
- Offsite/vault SMF retention may be suitable at SMFD DUMP time (HISTORY2 output DD by SMFD)

CA SMF Director implementation – SPLIT file example

– SCRT (IBM sub-capacity) reporting daily split file:

```
SPLIT ALL TODDD(IBMSCRT) SELECT(70 89)
  TIMEGDG(<hlq>.SMFDSCRT.DAILY) TIMEUNIT(SYSDA)
  TIMESPACE(CYL,100,100) TIME(2359) .
```

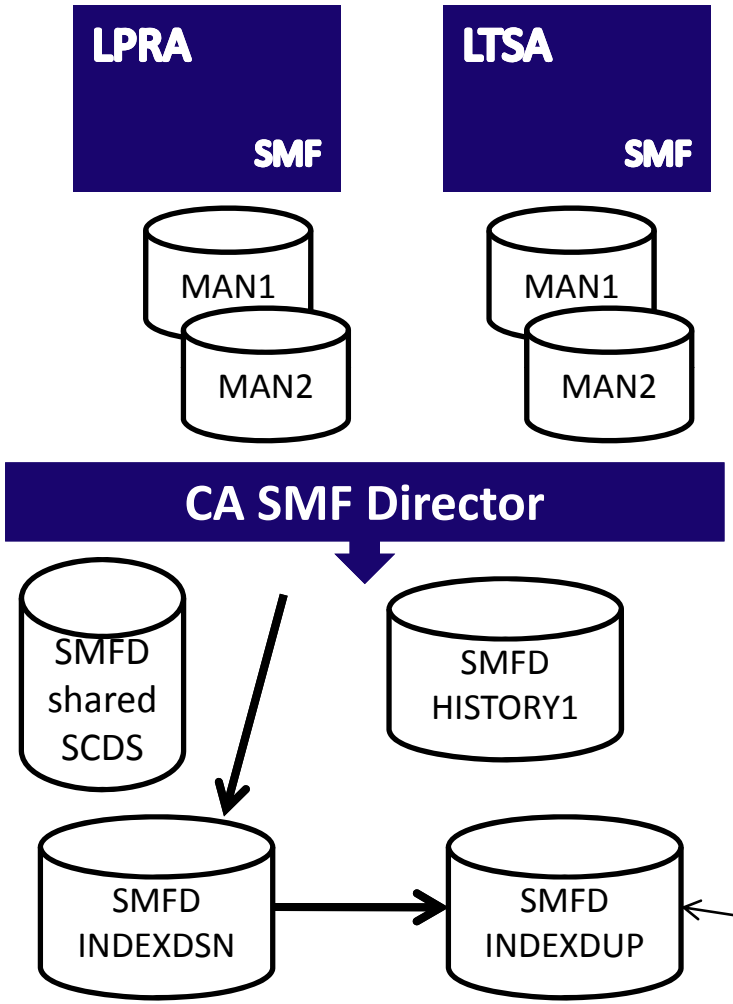
- INDEXDSN not required – no application interface here
- Setup GDG BASE for each LPAR, named <hlq>.SMFDSCRT.DAILY.SIDxxxx, where “xxxx” identifies the LPAR SMF ID.

– An SMFD EXTRACT to retrieve required SMF data on monthly basis also may satisfy the data requirement

CA SMF Director implementation – CA MICS interface overview

- Identify candidate MICS components: RMF, SMF, CICS, DB2
- One SPLIT file per component, unit, and LPAR, using TIMEGDG, TIME and INDEXDSN/INDEXDUP definitions using shared SCDS
- CA MICS update job INCRccc uses special DD for INDEXDUP dataset, DD name is SMFDCTR, for complete history files
- SMFD SPLIT file(s) processed on time-interval by CA MICS or other application
- CA MICS INCRccc job scheduled at suitable time-interval, just after “I SMF”, generated by auto-ops command

CA SMF Director implementation – CA MICS interface setup



SMFD SPLIT file – CA MICS interface parameters:

- `TIMEGDG(<hlq>.SMFDDB2.MICSINCR)`
- `TIME(0000,1200,1800)`
- `INDEXDSN(<hlq>.SMFDDB2.MICSINCR.PROD.INDX)`
- `INDEXDUP(MICXDB2)`

SPLIT File Index to track SMFD-generated split files:

- `<hlq>.SMFDDB2.MICSINCR.PROD.INDX`
- `<hlq>.SMFDDB2.MICSINCR.PROD.INDXDUP`

New SPLIT file created by SMFD with <u>new interval data</u>:

- `<hlq>.SMFDDB2.MICSINCR.SIDLPRA(+1)`
- `<hlq>.SMFDDB2.MICSINCR.SIDLTSA(+1)`

Run CA MICS INCRDB2 @ 12.30, 18.30, and CA MICS DAILY @ 00.30 – each reads INDEXDUP file

CA SMF Director implementation – CA MICS interface setup SPLIT file parameter definitions (continued)

– CA MICS components CICS, DB2, defined with three SPLIT file intervals:

```
SPLIT ALL TODD(MICSCIC) SELECT(110.001 110.002)
  TIMEGDG(<hlq>.SMFDCIC.MICSINCR)
  TIME(0000,1200,1800)
  TIMEUNIT(SYSDA) TIMESPACE(CYL,1000,2000)
  INDEXDSN(<hlq>.SMFDCIC.MICSINCR.PROD.INDX)
  INDEXDUP(MICXCIC) .
```

```
SPLIT ALL TODD(MICSDB2) SELECT(100 101)
  TIMEGDG(<hlq>.SMFDDB2.MICSINCR)
  TIME(0000,1200,1800)
  TIMEUNIT(SYSDA) TIMESPACE(CYL,1000,3000)
  INDEXDSN(<hlq>.SMFDDB2.MICSINCR.PROD.INDX)
  INDEXDUP(MICXDB2) .
```


summary

- CA SMF Director manages SMF data while reducing overhead with redundant data-capture and excessive SMF data handling
- SMFD, a key component with SMF Logstream migration
- The SMFD EXTRACT facility simplifies data requests, leveraging the SCDS (control dataset) for tracking where SMF data resides
- CA MICS provides an interface that can enhance CA MICS data availability with CA MICS Incremental Database Update and SMFD SPLIT file facility

q&a

please complete a session evaluation form

- The number for this session is **MR130SN**
- After completing your session evaluation form, place it in the basket at the back of the room

thank you