

IBM Education Assistance for z/OS V2R1

Item: HISSERV and Software Counters

Element/Component: BCP HIS



Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Appendix



Trademarks

- See url <http://www.ibm.com/legal/copytrade.shtml> for a list of trademarks.



Presentation Objectives

- Understand programming interface to access CPU Measurement Facility (CPU/MF) data in real time.
- Understand the Enhanced Monitor Facility and how HIS exploits it.
- Understand SMF Type 113 record updates



Overview

▪ Problem Statement / Need Addressed

- Users want a way to access the CPU/MF data. Current method must be one of:
 - Use HIS output files and SMF records, high latency
 - Use hardware facility directly, dangerous
- IBM wants to better understand customer system behavior
- SMF Type 113 subtype 2 format is confusing and difficult to process.
- Can't easily identify the specific machine associated with SMF Type 113 and .CNT file output.

▪ Solution

- Provide a new programming interface to allow safe and real time access to the CPU/MF data (HISSERV)
- Exploit the Enhanced Monitor Facility (EMF) on z196 to capture customer z/OS system behavior for IBM analysis (ZOS counters)
- New SMF Type 113 Subtype 1 record, changes to SMF Type 113 Subtype 2
- Update service to record machine's sequence code in SMF Type 113 and .CNT file

▪ Benefit / Value

- Greater and safer exploitation of CPU/MF data
- Better understanding of customer system behavior



The HISSERV Service

- As it has in the past, the MODIFY HIS,BEGIN command allows for the collection of instrumentation data and records it into Unix System Services (USS) files and SMF Type 113 Records
 - This, collectively, is now known as the HIS Profiler
 - The HIS Profiler uses the HISSERV service to collect its instrumentation data
 - NAME is **HISPROF**
- You can now use the HISSERV service to create your own profilers for customized collection and processing of instrumentation data.
 - Allows you to
 - Collect/process sampling data
 - Collect/process event (counter) data
 - React to certain system state-change events
 - More than one profiler can be active at the same time



The HISSERV Service (cont.)

- To collect instrumentation data an exploiter must register with the HISSERV service to declare its intentions. This registration identifies the exploiter as a profiler of the system.
 - A unique token identifies the profiler on subsequent HISSERV calls.
- A profiler of the system can:
 - Be notified via callback mechanism with sampling data
 - Query events (counters)
 - Be notified via callback mechanism of important profiling events (underlying system state has changed).
- A profiler of the system must explicitly declare its intention to stop profiling the system.
- Macros:
 - HISSERV: Executable macro to invoke service
 - HISYSERV: Mapping macro for answer areas/input areas/return codes
 - HISYEXIT: Mapping macro for generic exit routine parameter area
 - HISYSMPX: Mapping macro for sampling exit routine parameter area
 - HISYCTRS: Equates for counter definitions



The HISSERV Service (cont.)

- Sample types:
 - The only available sample types are those defined by CPU/MF.
 - See bitmask HisSmpTyp in macro HISYSERV.
 - Sampling controls (SAMPTYPE, SAMPFREQ, and BUFCNT) are globally enabled/disabled/managed by the operator via the MODIFY HIS command.
 - A profiler can only say “I want to have access to what is being collected”
 - Sampling data is passed back to the profiler as it becomes available.
- Event types:
 - can be thought of as “counter sets”.
 - are the counter sets defined by CPU/MF (Basic, Problem, Crypto, Extended) and the Enhanced Monitor Facility (ZOS).
 - See bitmask HisEvnTyp in macro HISYSERV
 - can be independently enabled/disabled by each profiler at registration time.
 - can be queried at any frequency by the profiler (snapshot of counter values).



The HISSERV Service – Interface

- Environment (PC routine):
 - Minimum Authorization: Supervisor state or PKM 0-7
 - Task mode
 - P=H=S
 - AMODE 31 or 64
 - Primary or AR ASC mode
 - Enabled for I/O and external interrupts
 - The caller must not be holding any locks
- Input Registers: None need to be set
- Output Registers:
 - 0: reason code if R15 nonzero
 - 1: used as work register by system
 - 2-13: unchanged
 - 14: used as work register by system
 - 15: return code



The HISSERV Service – Interface (cont.)

- HISSERV REQUEST=PROFILE,ACTION=START: Exploiter declares intention to start profiling the system.
 - Input
 - EVENT: Which event types exploiter intends to query
 - SAMPLE=YES|NO: Whether to be called back with sampling data
 - NAME: An externally identifiable name
 - EXITRTN: A module to get control for callbacks
 - Output
 - OUTPROFILETKN: The token to use for subsequent requests
- HISSERV REQUEST=PROFILE,ACTION=STOP: Exploiter declares intention to stop profiling the system
 - Input
 - PROFILETKN: The token received on the start request.



The HISSERV Service – Interface (cont.)

- HISSERV REQUEST=QUERY: Query service to obtain data. All queries require three parameters:
 - ANSAREA: Output area to hold the query results
 - ANSLEN: Length of storage supplied in ANSAREA
 - TYPE: The type of query
 - EVENT: Return data about the current event type state
 - SAMPLE: Return data about the current sample type state
 - PROFILERS: Return data about the current profiler state
 - EVENTDATA: Return event data
 - Requires a PROFILETKN to identify the profiler
 - Optional parameter CPUMASK can be used to query only a subset of the CPUs in the system.
- All ANSAREAs mapped by macro HISYSERV



The HISSERV Service – The Exit Routine

- The EXITRTN must be a load module in LPA, the LNKLIST concatenation, or the nucleus
- Parameter area mapped in macro HISYEXIT
- Entry Environment (documented in macro HISYEXIT):
 - Enabled, TCB mode, supervisor state, key 0, no locks held
 - H=P=S
 - AMODE 24/31/64 (same AMODE as linkedit attribute).
 - Registers:
 - 0
 - 1: Address to the input parameter area
 - 2-12: Irrelevant
 - 13: Address to a 216 byte save area
 - 14: Return address
 - 15: Exit routine entry point address
- Restrictions:
 - Cannot issue HISSERV REQUEST=PROFILE request within exit, or wait on a resource held by another work unit which might issue a HISSERV REQUEST=PROFILE.



The HISSERV Service – The Exit Routine (cont.)

- Two reasons the exit routine will be called:
 - A profiler needs to be notified of an underlying state change. Passed to the exit are:
 - Action being taken by the HISSERV service:
 - None: No disruption, just an FYI to the profiler
 - UpdateStart: The service needs to update itself, nothing can be done until...
 - UpdateDone: The service has finished updating itself
 - Reason for the notification (a bitmask -- it is possible for no bits to be set if the action being taken is None or UpdateDone):
 - An enabled CPU/MF Counter Set has become deauthorized
 - An enabled CPU/MF Sampling type has become deauthorized
 - CPU/MF counter data has been lost by the hardware
 - CPU/MF sampling data has been lost by the hardware
 - The speed of one or more processor types has changed
 - The service parameters have changed (MODIFY HIS command)
 - A profiler requested sampling data, and sampling data needs to be processed:
 - Macro HISYSMPX maps this data
 - Profilers get the same Sampling Data Buffers (SDBs) that are filled by the hardware.
 - Being quick about processing these is important, or overflow will occur.
 - Data should be treated as read-only by the profilers
 - Profilers need to understand the CPU Sampling Facility architecture in order to process these “raw” buffers.
- Macro HISYEXIT defines the interface to the exit routine.



The HISSERV Service – Processing Sampling Data

- General flow:
 - Use HISSERV REQUEST=PROFILE,ACTION=START to register an exit routine that will process the sampling data.
 - Exit routine is called when sampling data is available or an underlying system state change occurs.
 - Use HISSERV REQUEST=PROFILE,ACTION=STOP when it is time to stop the exit.
- Exit routines that involve processing of sampling data must be written with performance in mind.
 - Being quick about processing SDBs is important or overflow will occur.
 - Run-time analysis of the sampling data should *NOT* be done under the sampling exit.
 - IBM recommends copying the sampling data provided to the exit and doing additional processing / analysis under a different work unit.



The HISSERV Service – Processing Sampling Data (cont.)

- An exit routine that processes sampling data might look like:

```
Using HisExitParm,R1
```

```
CLI HisExitParm_Func,HisExitParmFunc_kStat
```

```
JNE CHECK_SMP
```

```
PUSH USING
```

```
USING HisStatParm,R1
```

* Called for a service action?

* No, see if “samples ready”

*

* Place code to process any service actions

*

```
J DONE
```

```
POP USING
```

```
CHECK_SMP DS 0H
```

```
CLI HisExitParm_Func,HisExitParmFunc_kSmp
```

```
JNE DONE
```

```
PUSH USING
```

```
USING HisSmpParm,R1
```

* Called for “samples ready”?

* No, we are done

*

* Place code to process the full SDBs

*

```
POP USING
```

```
DONE DS 0H
```

```
HISYEXIT
```

```
HISYSMPX
```



The HISSERV Service – Processing Event Data

- General flow:
 - Use HISSERV REQUEST=PROFILE,ACTION=START to register the intent to collect event data.
 - Possible event types: BASIC, PROBLEM, EXTENDED, CRYPTO, (and now) ZOS.
 - Which event types to collect controlled by bitmask passed in EVENT= parameter (see HisEvnTyp in macro HISYSERV for mapping)
 - Exit routine can be a no-op (IEFBR14) if you are not interested in processing sampling or system state callbacks.
 - Consider using SAMPLE=NO to limit callbacks to system state changes only.
 - Periodically use HISSERV REQUEST=QUERY,TYPE=EVENTDATA to get a snapshot of the counter sets.
 - Use HISSERV REQUEST=PROFILE,ACTION=STOP when you are done.
- Not all event types may be available to your profiler.
 - Use HISSERV REQUEST=QUERY,TYPE=EVENT to see what event types are available.



The HISSERV Service – Processing Event Data (cont.)

- An exploiter might do something like this:

myExit = << module name of exit routine (can be IEFBR14) >>

myName = << some 8 character name to associate with the profiler >>

myEvents = << Fullword with “Event Type” bits that the profiler is interested in turned ON >>

- * Start the profiler

```
HISSERV REQUEST=PROFILE,
```

```
    ACTION=START,
```

```
    EXITRTN=myExit,
```

```
    NAME=myName,
```

```
    SAMPLE=NO,
```

```
    EVENT=myEvents,      * could do something like EVENT==A(HisEvnTyp_ExtendedCtrls)
```

```
    OUTPROFILETKN=myTkn
```

- * Collect and process event data (every minute for 100 minutes)

```
DO X=1 TO 100;
```

```
    HISSERV REQUEST=QUERY,TYPE=EVENTDATA,PROFILETKN=myTkn,CPUMASK=ALL,
```

```
        ANSAREA=myAnsArea
```

```
    << Process Data >>
```

```
    << Wait 1 Minute >>
```

```
END;
```

- * Stop the profiler

```
HISSERV REQUEST=PROFILE,ACTION=STOP,PROFILETKN=myTkn
```



MODIFY HIS Command Changes

- New MODIFY HIS,SERVICE command allows changes to sampling parameters without having to start/stop a collection run
 - SAMPTYPE: Indicate which sampling types should be collected
 - SAMPFREQ: Indicate the sampling frequency samples are collected
 - BUFCNT: Indicate the number of sampling buffers per CPU
 - DDNAME: the referenced DD must only contain sampling parameters
- MODIFY HIS,SERVICE,REFRESH
 - Instructs the system to refresh the HISSERV service
 - Allows service to begin using counter sets/sample types recently authorized through HMC.
 - Mutually exclusive with all other MODIFY HIS,SERVICE parameters
- Changing the service parameters causes the service to update itself
 - Instrumentation data collection temporarily suspended for all profilers
 - Each profiler's exit routine receives an UpdateStart callback followed by an UpdateDone callback



MODIFY HIS Command Changes (cont.)

- All sampling parameters, either on MODIFY HIS,BEGIN or MODIFY HIS,SERVICE accept a new value: PERSIST
 - This means “keep the current setting”
 - Not specifying a parameter will reset it to its default
- PERSIST added to preserve compatibility for MODIFY HIS,BEGIN
 - For example: it is assumed anyone who issues MODIFY HIS,BEGIN without SAMPFREQ wants the default sampling frequency, not “whatever it currently is”.
 - Before, there was no concept of parameters that persisted across a HIS collection run.
 - To use the SAMPFREQ from the previous HIS collection run, “whatever it currently is”, use SAMPFREQ=PERSIST.
- Specify SAMPFREQ=PERSIST, SAMPTYPE=PERSIST, and BUFCNT=PERSIST on every MODIFY HIS,BEGIN if you don't want to change the sampling service parameters.
 - Consider using the DDNAME parameter to help make sampling service parameters consistent between runs.



MODIFY HIS Command Changes (cont.)

- CTRSET now collects either counters from hardware events (CPU/MF, which has little overhead) or software events (EMF, which has greater overhead)
 - CTRSET=ALL
 - **Deprecated, is now an alias for CTRSET=HARDWARE**
 - Did not want to include new software counters in this group for compatibility reasons; (avoid undesirable overhead in installations where CTRSET=ALL is being used).
 - CTRSET=HARDWARE
 - All hardware counter sets
 - Any new counter set which incurs minimal performance overhead will be added here
 - This is equivalent to CTRSET=ALL
 - CTRSET=SOFTWARE
 - All software counter sets (currently only the ZOS counter set).
 - CTRSET=COMPLETE
 - All counter sets: “I acknowledge there may be a performance overhead”
 - New counter sets may be added here



Displaying the state of the HISSERV service

- DISPLAY HIS is updated to show the HISSERV status (showing new section only):

HISSERV STATUS: ACTIVE|INACTIVE

EVENT

AUTHORIZED= NONE | [BASIC][,PROBLEM-STATE][,CRYPTO][,EXTENDED][,ZOS]

ENABLED= NONE | [BASIC][,PROBLEM-STATE][,CRYPTO][,EXTENDED] [,ZOS]

SAMPLE

AUTHORIZED= NONE | [BASIC][,DIAG]

ENABLED= NONE | [BASIC][,DIAG]

BUFCNT= xxxx (PAGES/PROCESSOR)

SAMPFREQ= xxxxxxxxx (SAMPLES/MINUTE)

PROFILER

NAME	START	QUERY	SAMPLE
XXXXXXXXX	hh:mm:ss.xxxxxxx	hh:mm:ss.xxxxxxx	xxxxxxx
XXXXXXXXX	hh:mm:ss.xxxxxxx	hh:mm:ss.xxxxxxx	xxxxxxx

- Note while the command status section shows what command was last issued for the HIS Profiler, this section shows what is currently in effect for the HISSERV service.
- Profiler query time is average time between queries.
- Profiler sample time is average number of microseconds it took to handle a sample callback.



The HIS.SERVSTAT Dynamic Exit

- Exit routines registered with the HIS.SERVSTAT dynamic exit will be called when HISSERV
 - is enabled (HIS address space coming up)
 - is disabled (HIS address space going down)
 - has started profiling the system
 - has stopped profiling the system
- Interface is the same as that for HISSERV exit routines.
 - Defined in macro HISYEXIT
- Documented in “MVS Installation Exits”



The ZOS Counter Set

- The Enhanced Monitor Facility (EMF) on z196 turns a monitor class into a “software driven counting facility”
- z/OS uses this to count specific software events. For example:
 - Number of times a particular SVC is issued
 - Number of times a task or SRB is dispatched
 - Number of times a particular lock is obtained
- HIS encapsulates these counts as the ZOS counter set
 - The HIS Profiler will write to the .CNT file and SMF Type 113 record, when requested via CTRSET=(ZOS).
 - Any profiler can request and query the ZOS counter set.
 - Currently, what each counter means is IBM internal only



ZOS Counter Data Format in .CNT File

- The ZOS counter data will be written into the current .CNT file on the USS filesystem when requested via CTRSET=(ZOS), CTRSET=SOFTWARE, or CTRSET=COMPLETE.
- Record in the .CNT file looks like:

```
COUNTER SET= ZOS
COUNTER IDENTIFIERS:
SOFTWARE COUNTER INFORMATION NOT AVAILABLE
```

```
START TIME: yyyy/mm/dd hh:mm:ss START TOD: xxxxxxxxxxxxxxxxx
END TIME:   yyyy/mm/dd hh:mm:ss END TOD:  xxxxxxxxxxxxxxxxx
EVENT COUNTERS (HEXADECIMAL) FOR CPU xx (CPU SPEED = x CYCLES/MIC):
0000-0003: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
0004-0007: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
0008-0011: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
0012-0015: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
0016-0019: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
0020-0023: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
...
1496-1499: xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx
```



SMF Type 113 Subtype 2 changes

- The SMF Type 113 mapping is defined in macro HISYSMFR.
- People find the subtype 2 record layout difficult to work with.
- Overview: Three main sections
 - SMF113_2_CTR: Main Section
 - SMF113_2_CSOF: Offset to counter set section
 - SMF113_2_CSON: Number of counter set sections
 - SMF113_2_CSLN: Length of counter set section
 - SMF113_2_CDOF: Offset to counter data section
 - SMF113_2_CDON: Number of counter data sections (total for all counter sets)
 - SMF113_2_CDLN: Length of counter data section
 - SMF113_2_CSS: Counter Set Section
 - SMF113_2_CST: Counter Set Type (Basic, Problem, Crypto, Extended)
 - SMF113_2_CSP: Bitmap identifying which counters in a counter set are available.
The idea is if a counter set only has counter 1 and 10 defined, space isn't wasted on nonexistent counters 2-9 (counter 1 and counter 10 are contiguous)
 - SMF113_2_CSN: The number of counter data sections for this counter set
 - SMF113_2_CDS: The Counter data section (8 bytes of data)



SMF Type 113 Subtype 2 changes (cont.)

- Example: You are interested in the 10th counter in the Extended Counter Set
(assume all counter and bit indices are 1-based for this example)
 - 1) Find offset to the start of the Extended Counter Set counters:
 - Set offsetToECSCounters to 0
 - Loop through SMF113_2_CSS sections:
Add the product of SMF113_2_CSN and SMF113_2_CDLN to offsetToECSCounters until Extended Counter Set section found.
 - The 1st counter in the Extended Counter Set starts at:
 $\text{SMF113_2_CDOF} + \text{offsetToECSCounters}$
 - 2) Check that the 10th counter is available by checking the 10th bit in SMF113_2_CSP.
 - If ON, proceed to the next step.
 - 3) Count the number of counters represented before the 10th counter
 - Count the number of ON bits between 1st and 9th bit in SMF113_2_CSP.
 - Set offsetToECSCounter10 to the product of “num bits on” and SMF113_2_CDLN.
 - 4) The 10th counter is at:
 $\text{SMF113_2_CDOF} + \text{offsetToECSCounters} + \text{offsetToECSCounter10}$



SMF Type 113 Subtype 2 changes (cont.)

- Turns out, there is no counter set that has discontinuous counters
 - The SMF113_2_CSP bitmask always has bit 1-n on, n being SMF113_2_CSN
 - Unnecessary complexity
- Same example, new method
 - 1) Exactly the same as before (unfortunately)
 - 2) $\text{OffsetToECSCounter10} = (9 * \text{SMF113_2_CDLN})$
 - 3) Extended Counter 10 is at $\text{SMF113_2_CDOF} + \text{offsetToECSCounters} + \text{offsetToECSCounter10}$
- IBM recommends all products processing SMF 113 subtype 2 records, from any hardware, stop using SMF113_2_CSP and use the new method described above.



SMF Type 113 Subtype 1

- Provides the same data as Subtype 2 (plus more)
 - Counter values are delta values (subtype 2 is absolute)
 - Can produce ZOS counter set counter data (not supported in subtype 2)
 - Counter data can be either 4 bytes or 8 bytes (subype 2 is always 8 bytes)
 - More intuitive record layout (hopefully)
- Four sections:
 - SMF113_1_CTR: Main Section
 - SMF113_1_CSOF: Offset to counter set sections
 - SMF113_1_CSLN: Length of counter set section
 - SMF113_1_CSON: Number of counter set sections
 - SMF113_1_CSS: Counter Set Section
 - SMF113_1_CSType: Counter Set Type (Basic, Problem, Crypto, Extended, zOS)
 - SMF113_1_CSUseLCDS: Bit, when on, use 8 byte counter (otherwise 4 byte counter)
 - SMF113_1_CDOF: Offset to counter data sections for this counter set
 - SMF113_1_CDLN: Length of counter data section
 - SMF113_1_CDON: Number of counter data sections for this counter set
 - SMF113_1_SCDS: Short Counter Data Section (4 bytes)
 - SMF113_1_LCDS: Long Counter Data Section (8 bytes)



SMF Type 113 Subtype 1 (cont.)

- Example: You want counter 10 in the extended counter set
 - 1) OffsetToECSCounters = SMF113_1_CDOF in the Extended Counter Set Section
 - 2) OffsetToECSCounter10 = (9*SMF113_1_CDLN)
 - 3) Extended Counter 10 is at offsetToECSCounters + offsetToECSCounter10

- Other differences:
 - Subtype 2 has bits for first/middle/last records, subtype 1 does not.
 - The first subtype 2 record is at the very start of the collection run, the first subtype 1 record is the first interval.
 - The delta value must have a base

- **Future enhancements will be made only to SMF 113 subtype 1 records.**



Machine sequence code recording

- New fields in SMF type 113 subtype 1 and 2
 - Records the sequence code of the machine
 - SMF113_1_MachSeqCode
 - SMF113_2_MachSeqCode

- Sequence code added to header of .CNT file:

```
HIS019I EVENT COUNTERS INFORMATION VERSION 4
FILE NAME: SYSHIS20130226.113831.000.CNT
COMMAND: MODIFY HIS,B
LOSS OF SAMPLE DATA ALERT: NO    SAMPLE BUFFER OVERFLOW COUNT: 0
LOSS OF COUNTER DATA ALERT: NO
STATE CHANGE: NO
MODEL: 2827-743  SEQCODE: 0000000000035DC7
```



Interactions & Dependencies

- Software Dependencies
 - None
- Hardware Dependencies
 - z10 or later hardware for CPU/MF
 - z196 or later hardware for EMF
- Exploiters
 - None



Migration & Coexistence Considerations

- SMF Type 113
 - In SMFPRMxx, specifying TYPE(113) will give you SMF Type 113 Subtypes 1 and 2.
 - If this is not what you want, specify the desired subtype.



Installation

- The HISSERV service and the HIS Profiler are only available when the HIS address space has been started.



Appendix

▪ Publications:

- MVS Authorized Assembler Services Reference, Volume 2 (EDT-IXG)
[SA23-1373]
- MVS Installation Exits [SA22-7593]
- MVS System Commands [SA38-0666]
- MVS System Management Facilities (SMF) [SA38-0667]
- MVS System Messages Volume 6 (GOS-IEA) [SA38-0673]

