IBM Education Assistance for z/OS V2R2

Item: SDUMP Usertoken Support
Element/Component: BCP Service Aids

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Installation
- Presentation Summary
- Appendix

Trademarks

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

Presentation Objectives

- Learn about DUMPABLE attributes and SDATA
- Learn about how new SDATA options can be used to limit the high virtual (HV) common storage in an SVC dump (SDUMP)
- Learn about usertoken and how to exploit it to customize selection of HV storage in an SDUMP

Overview

- Problem Statement / Need Addressed
  - Increased SDUMP capture times
    - Gigabytes of LIKECSA storage, which has no significance to the problem being diagnosed
  - Impacts media storage and data transmission times
    - Larger SDUMPs

- Solution
  - (Tactical) Provide options to limit the LIKECSA storage in SDUMPs
  - (Strategic) Support requesting HV storage using the memory/user token

- Benefit / Value
  - Reduces the size of SDUMPs
  - Improves SDUMP capture and transmission times
  - Simpler storage selection algorithm for HV storage

Usage & Invocation - Background: SDATA option

- Used to request the storage areas to include in a dump
  - CSA, SQA, LSQA, RGN, ...
- Supported by SDUMP, SYSMDUMP, SYSABEND and SYSUDUMP
- Supported on
  - System commands CHNGDUMP, DUMP, SLIP
  - Parmlib members IEADMCxx (SDUMP), IEADMRxx (SYSMDUMP), IEAABDxx (SYSABEND) and IEADMPxx (SYSUDUMP)

Usage & Invocation - Background: DUMPABLE attribute

- What is a dumpable attribute and how to define it?
  - Available for HV storage at memory object creation time
  - RSM services allow to define storage's dumpable attributes via DUMP=
    - Requested explicitly or defaulted
    - HV common storage can have LIKECSA or LIKESQA
    - HV private storage can have LIKELSQA or LIKERGN
    - HV storage with no dumpable attributes will not be dumped
      - DUMP=NO | SVCDUMPRGN=NO

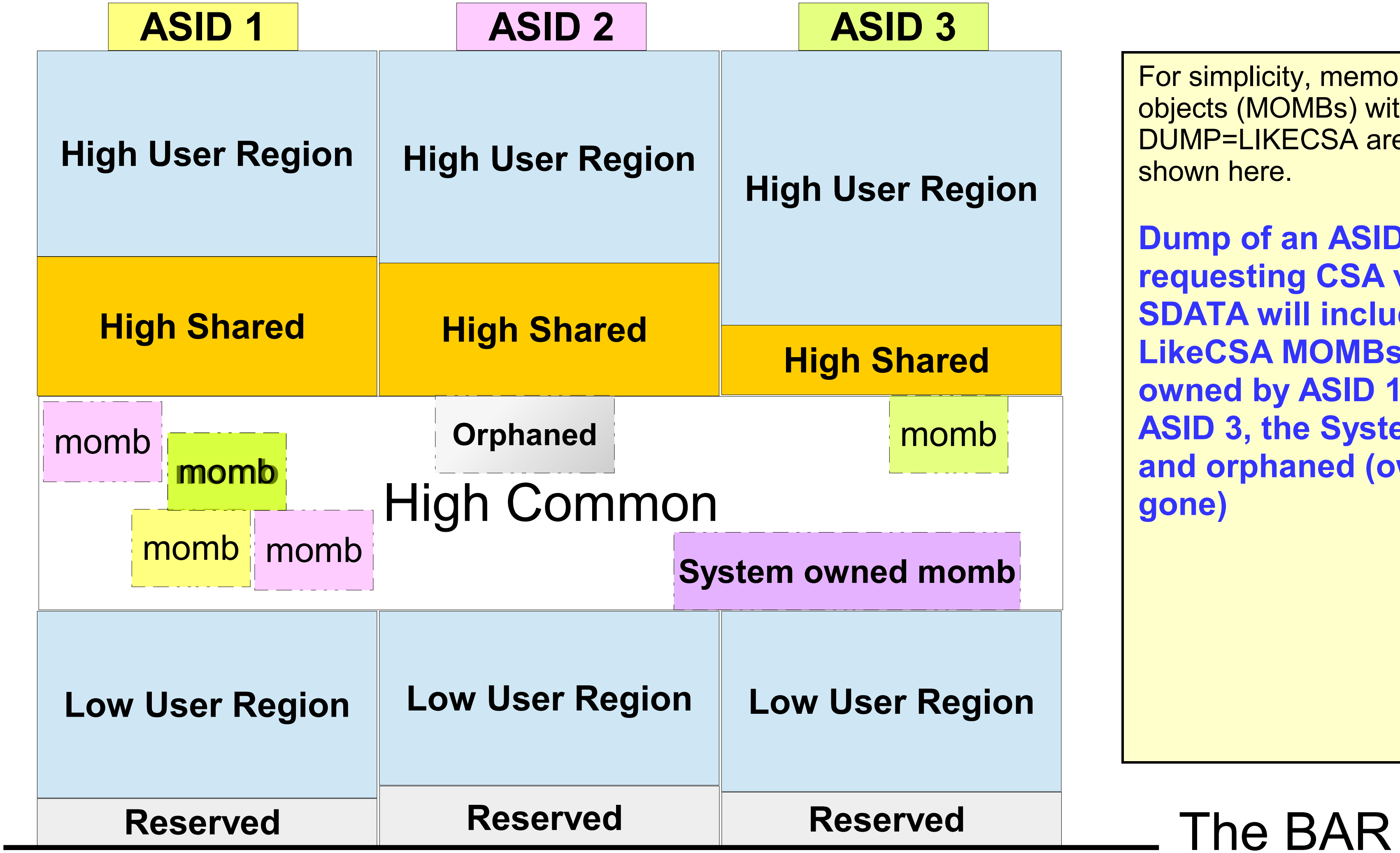
Usage & Invocation - Background: How to include HV storage in a dump

- SDATA= SQA includes SQA, ESQA and HV common storage with DUMP=LIKESQA attributes
- SDATA=CSA includes CSA, ECSA and HV common storage with DUMP=LIKECSA attributes
- SDATA=RGN includes user region, extended user region and HV private storage with DUMP=LIKERGN attributes
- SDATA=LSQA includes LSQA, ELSQA and HV region storage with DUMP=LIKELSQA attributes
  - LSQA is a subset of user region storage
- Use LIST64 or SUMLIST64 to dump HV storage without dumpable attributes

Usage & Invocation – Background: How to include HV storage in a dump.. continued

- LIKECSA storage is an easy way to share data
  - Other methods available, but require overhead
    - HVSHARE solution requires complex setup
  - SDATA=CSA will cause inclusion of all LIKECSA storage along with CSA and ECSA
    - LIKECSA storage owned by other ASIDs, SYSTEM or orphaned may not be needed for diagnostic purposes
- ALL or NOTHING approach

Background – How is DUMP=LIKECSA storage dumped?



For simplicity, memory objects (MOMBs) with DUMP=LIKECSA are shown here.

Dump of an ASID 2 requesting CSA via SDATA will include LikeCSA MOMBs owned by ASID 1, ASID 3, the System and orphaned (owner gone)

Usage & Invocation – HCSA solution overview

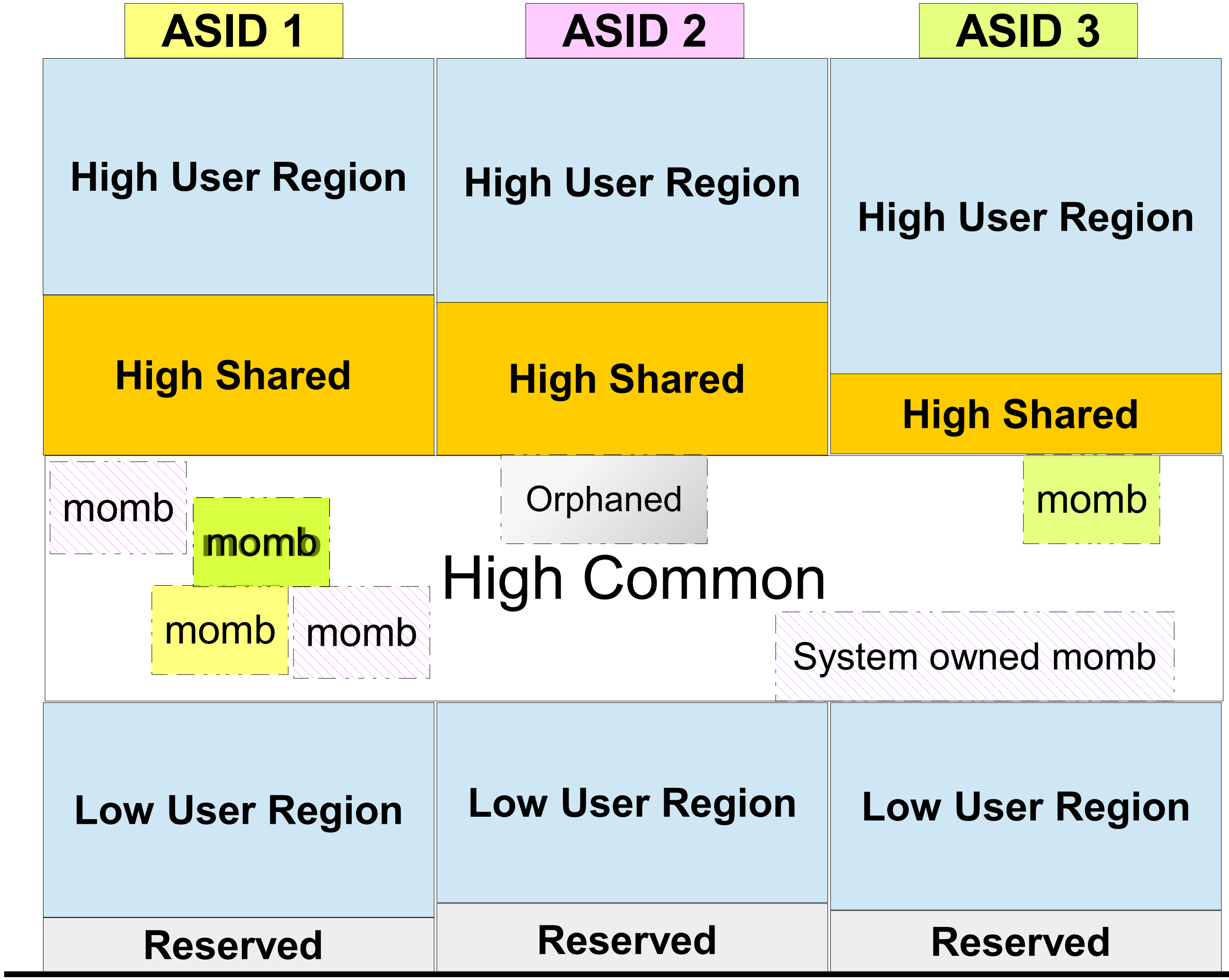
- Limit the LIKECSA storage in a dump
  - Dump LIKECSA storage based on the ownership

- SDUMP to support 3 new SDATA *filter* options
  - HCSAByASID – LIKECSA storage owned by the ASIDs that are being dumped by SDUMP
  - HCSASysOwner – LIKECSA storage owned by the SYSTEM
  - HCSANoOwner – LIKECSA storage no longer owned by an active ASID

  - Will not affect dumping of any below the bar storage
  - Will not affect dumping of LIKESQA storage

- HV common is not divided
  - HCSA storage means HV common storage with DUMP=LIKECSA dumpable attributes in this presentation

Solution Overview – LIKECSA storage dumped w/new options



For simplicity, memory objects (MOMBs) with DUMP=LIKECSA are shown here.

SVC Dump of an ASID 2 requesting SDATA of HCSAByASID and HCSASysOwner will NOT include LikeCSA MOMBs owned by ASID 1, ASID 3, or orphaned (owner gone)

*HCSA filters only affect dumping of HV common with DUMP=LIKECSA attributes.

Usage & Invocation - HCSA solution continued

- New SDATA options
  - Supported for SDUMP
  - Available on all interfaces of SDUMP
    - SDUMPX Service
    - CHNGDUMP, DUMP, and SLIP SET, A=SVCD|SYNCSVCD System commands
    - IEADMCxx parmlib member
  - Allowable abbreviations
    - HCAS for HCSAByASID
    - HCSY for HCSASysOwner
    - HCNO for HCSANoOwner
    - Note that DUMP command supports only abbreviated names
  - Recommended usage as a system filter on command interface

Usage & Invocation - HCSA solution continued

- SDATA options allowed for all SDUMP external interfaces

,SDATA=(sdata options),	Sdata options=Any combination of the following, separated by commas: ALLNUC, ALLPSA, COUPLE, CSA, GRSQ, **HCSAByASID, HCSANoOwner, HCSASysOwner,** LPA, LSQA, NOALLPSA/NOALL, NOSQA, NOSUMDUMP/NOSUM, NUC, PSA, RGN, SERVERS, SQA, SUMDUMP/SUM, SWA, TRT, XESDATA, DEFAULTS/DEFS, NODEFAULTS/NODEFS, IO

Usage & Invocation - HCSA solution continued

- CHNGDUMP Command
  - Change the mode and dump options list for SDUMP, SYSABEND, SYSMDUMP and SYSUDUMP
  - Filters introduced to assist an installation to control the SDUMP size
    - HCSAByASID
    - HCSANoOwner
    - HCSASysOwner
  - Use these filters cautiously
    - See the table on the next page for details

Usage & Invocation - HCSA solution continued

- The following table describes how these new high CSA SDATA options affect the CSA storage capture in an SDUMP:

Specified SDATA option or options	**CSA storage that is included in the dump**
CSA	All above the bar and below the bar CSA storage.
CSA, HCSAByASID, HCSANoOwner, HCSASysOwner	All below the bar CSA storage, high virtual CSA storage that is owned by the ASIDs included in the dump, high virtual CSA storage that is no longer owned by an active ASID, and high virtual CSA storage that belongs to the SYSTEM. The dump does not include high virtual CSA storage that is owned by the ASIDs that are excluded from the dump.
HCSAByASID, HCSANoOwner, HCSASysOwner	All high virtual CSA storage that is owned by the ASIDs included in the dump, high virtual CSA storage that is no longer owned by an active ASID, and high virtual CSA storage that belongs to the SYSTEM. No below the bar CSA storage is included in the dump.
(neither CSA nor any of the HCSAxxxx options)	None of the CSA storage is included in the dump.

Usage & Invocation – Background: Memory/user token

- What is a memory/user token and how to create it?
  - Available for HV storage at memory object creation time
  - RSM services allow a memory object to be associated with a token
  - Users can provide an 8 byte token (usertoken) or request the System to create one (memory token)
    - MOTKNSOURCE=user|system
  - One or more memory objects can be associated with one memory/user token

NOTE: Both system created memory token and user supplied usertoken are referenced as usertoken in this presentation.

Usage & Invocation – Usertoken Solution

- Solution overview
  - Provides a selection algorithm to dump HV storage in an SDUMP
    - Dump HV storage using a usertoken
    - Memory objects must be created with a token and no dumpable attributes (DUMP=NO or SVCDUMPRGN=NO)
  - SDUMP to support a new keyword, UTOKEN= on SDUMPX interface
    - Can be used for HV common and region storage

Usage & Invocation - Usertoken solution continued

- Usertoken area
  - Must reside below the bar
    - ABEND233 RSND0 if UTOKEN area is not accessible

  - If exceeds 512 bytes
    - It must be in common storage
    - Must be available until data capture is complete
      - For asynchronous SDUMP, request must use ECB or SRB parameter to be notified when the data capture is complete

Usage & Invocation - Usertoken solution continued

- Usertoken area consists of
  - 8 bytes header
    - First 4 bytes indicate the size of usertoken area, followed by 4 reserved bytes
  - 12 bytes usertoken entries (uentries)
    - 2 bytes ASID – relevant for HV region storage
    - 1 char Storage_type
      - 'C' or 'c' for HV common storage
      - 'R' or 'r' for HV region storage
    - 1 byte reserved
    - 8 bytes usertoken – used at memory object creation

- Maximum of 256 usertoken entries will be processed by SDUMP
  - Max of 3080 bytes of Utoken area

Usage & Invocation - Usertoken solution continued

- Must provide at least one full usertoken entry

- For usertoken entry
  - ASID can be omitted for
    - An HV common
    - An HV common + HV region for all participating ASIDs
  - ASID must be one of the address spaces included in the dump

Usage & Invocation - Usertoken solution continued

- The following table describes effects of usertoken entry input on the data captured in an SDUMP

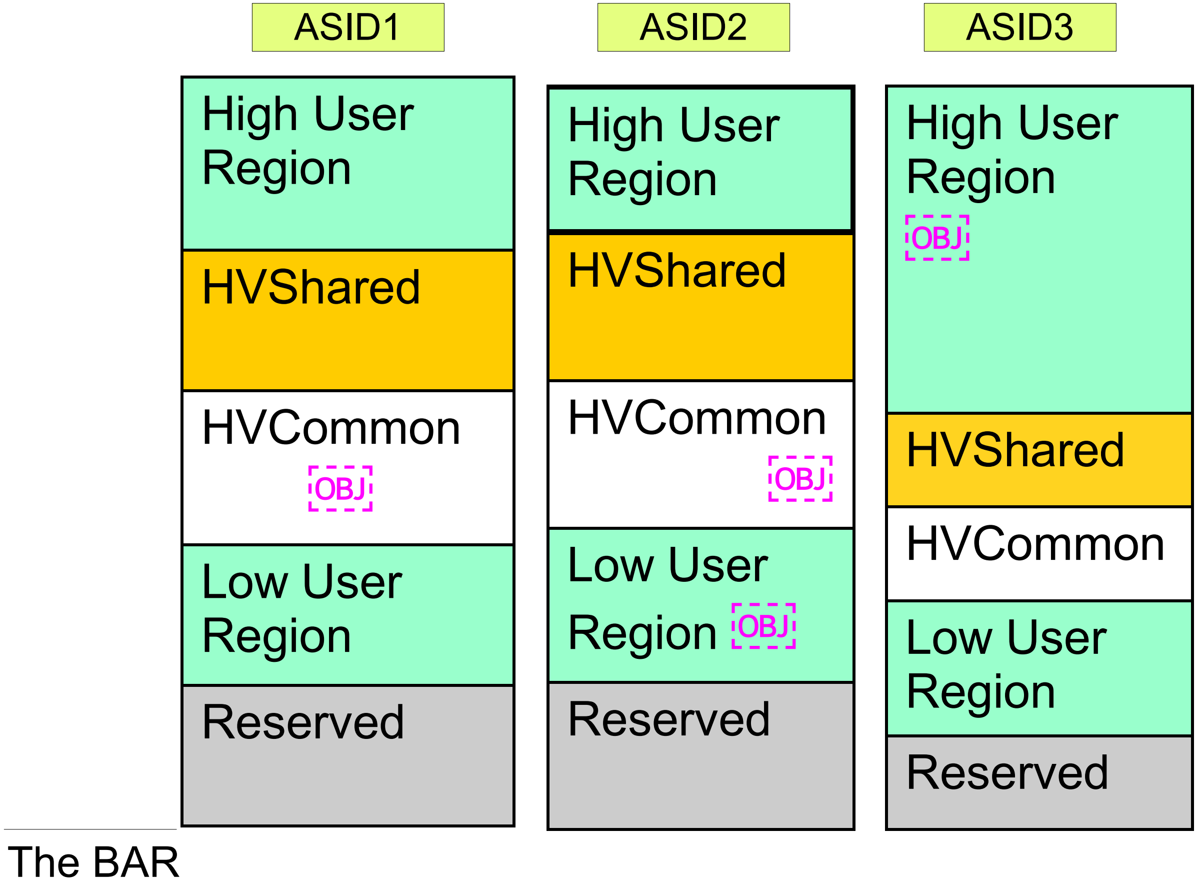
Storage_type	ASID	Usertoken	Matching HV Storage dumped
'C/c'	-	Y	HV Common
'C/c'	Y	Y	HV Common (ASID ignored)
-	Y	Y	HV Common + HV RGN for specified ASID
-	-	Y	HV Common + HV RGN for ALL dumping ASIDs
'R/r'	Y	Y	HV RGN for specified ASID
'R/r'	-	Y	HV RGN for all ASIDs being dumped
Y	Y	-	--

'-' means not specified, Y means proper input is specified

Usage & Invocation – Usertoken solution continued

- IBM strongly recommends exploiting HV storage usertoken usage
  - Use usertokens instead of dumpable attributes
    - Makes storage dumpable without being associated with larger dumpable attributes (LIKECSA, LIKESQA, LIKERGN..)
  - Not required, but create different usertokens for storage that has different diagnostic characteristics
    - One usertoken for all common storage
    - One usertoken for all region storage

Usage & Invocation - Usertoken solution continued



Same usertoken is associated with all 4 Memory Objects (MOMBs)

Utoken entry ('C',00,token) will dump 2 HVCommon MOMBs

Utoken entry ('C',02,token) will dump 2 HVCommon MOMBs (ASID ignored)

Utoken entry('R',02,token) will dump ASID2's HVRegion MOMB

Utoken entry('R',-,token) will dump HVRegion MOMBs from ASID2 and ASID3

Utoken entry('R',01,token) will NOT dump any MOMB

Utoken entry(-,-,token) will dump all 4 MOMBs

Usage & Invocation - HCSA example

- Exploit new SDATA option via IEADMCxx

To request the dump of TCP/IP on this system, including only the high virtual CSA that is owned by the ASID and a summary dump, using a DUMP command parmlib member:

1. Create member IEADMCTP that contains the following:

```
JOBNAME=TCPIP, SDATA=(SUM,HCAS)
```

2. Run the following command on the MVS console

```
DUMP TITLE=(TCPIP dump), PARMLIB=TP
```

Usage & Invocation - HCSA example

- Exploit new SDATA option via DUMP command

```
DUMP COMM=(DUMP HCAS +CSA-TASK-WTOR)

0006 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

R 0006,SDATA=(CSA,SQA,HCAS,HCSY),END

IEE600I REPLY TO 0006 IS;SDATA=(CSA,SQA,HCAS, HCSY),END

IEA045I AN SVC DUMP HAS STARTED AT TIME=16.50.08 DATE=06/17/2014

FOR ASID (0001)

QUIESCE = YES

IEA794I SVC DUMP HAS CAPTURED:

....
```

Usage & Invocation - HCSA example

- Set/Display a SLIP requesting new SDATA options

```
SY1   SLIP Set,SBT,PVTMOD=(HVCSA,104,11C),A=(SVCD),sdata=(hcas,hcno,lsqa),
 id=SLP1,end
SY1   IEE727I SLIP TRAP ID=SLP1 SET
SY1   d slip=slp1
SY1   IEE735I 11.34.39 SLIP DISPLAY 603                                  C
ID=SLP1,PER-SBT,ENABLED(INACTIVE)
ACTION=SVCD,SET BY CONS C3E0SY1,PRCNTLIM=10,00,MATCHLIM=1,0
PVTMOD=HVCSA,00000104,0000011C
SDATA=(SQA,LSQA,HCSABYASID,HCSANOOWNER)
```

Usage & Invocation - HCSA example

- Set new SDATA options via CHNGDUMP command

```
SY1   cd set,sdump=(hcsy,hcsabyasid)

SY1   IEE712I CHNGDUMP PROCESSING COMPLETE

SY1   d d,o

SY1   IEE857I 11.11.00 DUMP OPTION 555

  SYSABEND- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM),

                      PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)

  SYSUDUMP- ADD PARMLIB OPTIONS SDATA=(SUM), NO PDATA OPTIONS

  SYSMDUMP- ADD PARMLIB OPTIONS (NUC,SQA,LSQA,SWA,TRT,RGN,SUM)

  SDUMP- ADD OPTIONS (HCSABYASID,HCSASYSOWNER,XESDATA),

                    BUFFERS=00000000K,MAXSPACE=00000500M,

                    MSGTIME=99999 MINUTES,MAXSNDSP=015 SECONDS,

                    AUXMGT=ON ,DEFERTND=NO

  ABDUMP- IGNORE DUMP REQUESTS
```

Usage & Invocation - HCSA example

- Delete new SDATA option via CHNGDUMP command

```
SY1   SY1   cd del,sdump=hcsy

SY1   IEE712I CHNGDUMP PROCESSING COMPLETE

SY1   d d,o

SY1   IEE857I 11.16.05 DUMP OPTION 561

  SYSABEND- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM),

                       PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)

  SYSUDUMP- ADD PARMLIB OPTIONS SDATA=(SUM), NO PDATA OPTIONS

  SYSMDUMP- ADD PARMLIB OPTIONS (NUC,SQA,LSQA,SWA,TRT,RGN,SUM)

  SDUMP- ADD OPTIONS (HCSABYASID,XESDATA),BUFFERS=00000000K,

                      MAXSPACE=00000500M,MSGTIME=99999 MINUTES,

                      MAXSNDSP=015 SECONDS,AUXMGMT=ON ,DEFERTND=NO

  ABDUMP- IGNORE DUMP REQUESTS
```

Usage & Invocation - HCSA example

- View new options in formatted SDUMP parmlist

```
= > IP CBF rtct+9c? STR(SDUMP) VIEW(X'FFFF')

SDUMP_PL: 027C3F48

   +0000  FLAG0.... 1C        FLAG1.... A1        SDATA.... 1402

   +0004  DCBAD.... 00000000  STORA.... 00000000  HDRAD.... 027C3840

   +0010  ECBAD.... 07CC989C  SRBAD.... 07CC989C  CASID.... 0001

   +0016  TASID.... 0001      ASIDP.... 00000000  SUMLP.... 00000000

   +0020  SDDAT.... 00000000  FLAG2.... 00        CNTL1.... C0

   +002A  TYP1..... 10        VERSN.... 03        SDTA2.... 25506000

   +002C  EXIT..... 2550      SDAT3.... 60     SDAT4.... 00

   +0030  SPLST.... 00000000  KYLST.... 00000000  RGPSA.... 00000000

   +003C  DCBA..... 00000000  STRAL.... 00000000  HDRA..... 00000000

   +0048  ASDLA.... 00000000  SMLA..... 00000000  SBPLA.... 00000000

   +0054  KEYLA.... 00000000  LSTDP.... 00000000  LSTDA.... 00000000

   .....
```

Usage & Invocation - HCSA example

- View new options in formatted SDUMP parmlist..continued

```
==> FLAGS SET IN SDUFLAG1:

 SVC dump request.

 48+ byte parameter list.

==> FLAGS SET IN SDUSDATA:

 Dump SQA.

 Dump rgn-private area.

 Dump all nucleus.

. . .

==> FLAGS SET IN SDUSDAT3:

 IO specified.

 Dump LIKECSA by ASID.

==> FLAGS SET IN SDUFLAG3:

 SDUMP was invoked by DUMP command.
```

Usage & Invocation - Usertoken example 1 (pseudo)

Request HV common storage via UTOKEN=

- Associate a memory object with a token

```
MyToken = 'DmpCmn01'

IARV64 Request=GETCOMMON,MoTknSource=User,MoTkn=MyToken,Dump=No
```

- Prepare usertoken area to dump HV common

```
utknlist = ''
utknlen = 8+12
utknstor(1) = 'C'
utknasid(1) = '001C'x
utknname(1) = MyToken
```

- Pass UTOKEN= on SDUMPX

```
SDUMPX HDR='UTKNCOM1 TESTCASE',SDATA=(ALLNUC,SQA,TRT,SUM),
       UTOKEN=utknlist
```

Usage & Invocation - Usertoken example 2 (pseudo)

Request HV common (system created token) and HV region (user created token) for all dumping ASIDs

- Associate memory objects with tokens

```
IARV64 Request=GETCOMMON,MoTknSource=System,OutTkn=SysToken,Dump=No

MyToken = 'DmpRgn01'

IARV64 Request=GETSTOR,MoTknSource=User,MoTkn=MyToken,Dump=No
```

- Prepare usertoken area to create 2 usertoken entries

```
utknlist = ''
utknlen = 8+(2*12)
utknstor(1) = 'C'
utknname(1) = SysToken
utknstor(2) = 'r'
utknname(2) = MyToken
```

- Pass UTOKEN on SDUMPX

```
SDUMPX HDR='UTKNCOM1 TESTCASE',SDATA=(ALLNUC,SQA,TRT,SUM),ASIDLST=ASIDList,
        UTOKEN=utknlist
```

Usage & Invocation – Messages

```
SY1   dump comm=(FP0173 scenario x)

SY1 *06 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

SY1   6,sdata=(hcsa,rgn),e

SY1   IEE600I REPLY TO 06 IS;SDATA=(HCSA,RGN),E

SY1   ASA101I SYNTAX ERROR: <NON-KEYWORD> WAS SEEN, WHERE ONE OF

(ALLNUC COUPLE CSA HCAS

HCNO HCSY GRSQ LPA

LSQA NOPSA NOSQA NOSUM

NUC PSA RGN SERVERS

SQA ....)

WOULD BE CORRECT.

DETECTING MODULE IS IEAVTS0L

SY1   IEE711I SYSTEM DUMP NOT TAKEN. DUMP SPECIFICATION NOT VALID
```

Installation

- Keep in mind:
  - New SDATA options specified via CHNGDUMP in ADD mode will affect high CSA storage collection for any SVC dump taken on the System

Presentation Summary

- Now you should be able to:
  - Utilize new HCSA SDATA options to limit storage in SDUMPs
  - Exploit usertoken(s) to customize HV storage in SDUMPs

Appendix

- Publications:
  - z/OS MVS System Commands, SA38-0666
  - z/OS MVS Authorized Assembler Services Reference, Vol 3, SA23-1374
  - z/OS MVS Diagnosis: Tools and Services Aids, GA32-0905
  - z/OS MVS Initialization and Tuning Reference, SA23-1380
  - z/OS MVS System Codes, SA38-0665
  - z/OS MVS Programming: Assembler Services Reference, Volume 2 (IAR -XCT), SA23-1370