



IBM IT Education Services

# E58

Charles E. Olsen

Problem Determination for VSE/ESA

**VSE Technical Conference**

November 10 - 12, 2003 | Hilton, Las Vegas, NV

© 2003 IBM Corporation

# Problem Determination for VSE/ESA

## Session E58

### Analysis of Common Problems

- Tracing Hardware Errors..... Page 4
- Tracing GETVIS Shortage.....Page 23
- VSAM File Access Conflicts.....Page 25
- Loop during CICS Initialization .....Page 31
- Correlate Dump with COBOL Source .....Page 37
- Stand-alone Dump to Disk .....Page 49
- Sharing Physical Tape Units .....Page 61
- Installing Service from Disk .....Page 63

VSE/ESA Technical Conf  
Las Vegas  
Nov 10th -12th , 2003

Charles E. Olsen  
[colsen@us.ibm.com](mailto:colsen@us.ibm.com)

# Problem Determination for VSE/ESA

## Session E58

### PD/PSI Console Commands:..... Page 65

- ALTER / DSPLY
- CANCEL cuu,FORCE
- DTRIATTN
- DEBUG
- DIAG
- DOIO
- FLT
- GETVIS
- IESINSRT
- LOCATE
- MAP
- MTC
- NOASSGN
- NOAVR
- NOUNL
- ONLINE cuu,FORCE
- QUERY DSPACE
- READY cuu
- PAUSE / SUSPEND / RESUME
- REIPL / PWROFF
- SHOW
- SIR
- STACK
- STATUS
- System Level
- TAPE
- TIME
- TERSE
- TRACE
- VOLUME (tape)

Some of these enhancements require the following VSE/ESA Service:  
VSE/ESA 2.6: UD52408    VSE/ESA 2.7: UD52409

# Tracing Hardware Errors

## Symptom:

```
BG 0001 1Q47I BG JACDITCT 33432 FROM VSEPROD(K00B) U=' 125000a'  
BG // JOB JACDITCT JOB TO LOAD CARD TO TAPE  
BG DITTO OUTPUT=SYS030, VOLSER=TAPTST, OUTFILE='TAPETEST'  
BG DITTO ASCII=NO  
BG 0014 OP20I P RECOVY ERR SYS030=180  
      CCSW=8B100000000E000000 CCB=566EF4  
      SNS= 0844042C 04422D00 00080000 0009E406 29917000 001A00  
BG JOB TERMINATED DUE TO RETURN CODE  
BG EQJ JACDITCT MAX. RETURN CODE=0016  
      DATE 02/23/2000, CLOCK 11/54/44, DURATION 00/00/05  
BG 0001 1Q34I BG WAITING FOR WORK
```

## Resolution:

Hardware CE used Sense data to find problem

## Tracing Hardware Errors

### Example 2:

```
FA 0014 OP64I R MAINT REQD SYS093=604
CCSW=6330D7588802000000 CCB=D75840 SK=000080000000
SNS= 00003000 2837CF0A ED008002 09AA4004 23000BE3 0001E3
05100200 AE282900
```

### Resolution:

Hardware warning message

# Tracing Hardware Errors

## Speaker Notes:

Customer is having a problem writing 1600 BPI IRS and BANK tapes. No software changes have been made recently. He has upgraded from a 9021 to a 7060H30. When an assign is typed in for a tape drive and followed with a mode of C0, he receives following message. Will not write in C0 mode.

```
OP20I-P Recovy Err ccsw=8B10000000E00000.
```

CCW Opcode

Failing CSW

I/O Response

## Resolution:

*"We should close this problem. The hardware repairman used the SDAID dump to locate a failed piece of the controller for the 3420 tape drives. After replacing the part the problem no longer existed."*

## Another example, this time from VSE-L newsgroup:

```
FA 0014 OP64I R MAINT REQD SYS093=604
CCSW=6330D7588802000000 CCB=D75840 SK=000080000000
SNS= 00003000 2837CF0A ED008002 09AA4004 23000BE3 0001E35B
05100200 AE282900
```

## Resolution:

*"I just got off the phone with IBM. They asked me if we were using a spare drawer, to which I was able to give my favorite answer, "Um...what's that?" But, it's OK because the controller had already phoned home and told IBM that one of the drawers had a bad battery. IBM will replace it on Monday and while the CE is here, I'll find out more about this spare-drawer business."*

# Tracing Hardware Errors

## Analysis Procedure:

1. Initiate SDAID tracing:

```
// EXEC SDAID  
OUTDEV T=<cuu1>  
TRACE IO AREA=ALL UNIT=<cuu2> OUTPUT=(TOD, CCWD=64)  
TRACE SIO AREA=ALL UNIT=<cuu2> OUTPUT=(TOD, CCWD=64)  
/*
```

2. Mount a scratch tape for SDAID output on <cuu1>.

3. Set up the failing condition.

4. Start SDAID tracing with “STARTSD” (or “STRTSD”).

5. Reproduce the problem

6. Terminate SDAID tracing issuing “ENDSD”.

7. Print the SDAID tape using “DOSVSDMP” (option 4).

8. Show trace to hardware support person.

# Tracing Hardware Errors

## Speaker Notes:

Most messages from 0P00 – 0P90I are triggered by a hardware response (or lack of response) to a software initiated I/O request. Unless the cause is inherently obvious, it is helpful to obtain an SDAID I/O trace, which will show the sequence of commands being issued, and the hardware response. Follow these steps:

1. SDAID tracing may be initiated either by executing a VSE job (in any partition), or by entering the commands (starting with “SDAID”) from the console. <cuu1> is the scratch tape for SDAID output, and <cuu2> is the tape whose I/O is to be traced.
- 4/5. “**STARTSD**” (or “**STRTSD**”) and “**ENDSD**” commands are attention routine (AR) commands, and can be entered at any console. “**ENDSD**” should rewind the SDAID trace output tape.
7. Print the SDAID tape using “**DOSVSDMP**” (option 4). This can be executed from the console via a PAUSExx job, or via the following JCL:

```
// JOB PRINT SDAID TAPE  
// EXEC DOSVSDMP, PARM='PRINT SDAID TAPE=<cuu1> FILE=1'  
/*  
/&
```

8. Locate the SDAID trace entry whose CSW matches what is shown in the console error message. The following I/O command will be a “sense” CCW, and the output can be interpreted from the appropriate hardware device reference manual.



# Tracing Hardware Errors

## Potential Problems using SDAID:

1. Turbo Dispatcher considerations
2. SDAID uses the SDAREA as work space.
3. Wait state due to output media problems:
  - a) Replace the tape (if required)
  - b) Make the tape unit ready
  - c) Press the external interrupt key

## VTAM SNA:

1. Trace address of the controller.  
  
-or-
2. use VTAM Generalized Trace Facility (GTF).

# Tracing Hardware Errors

## Speaker Notes:

### Potential Problems using SDAID:

1. Starting with VSE/ESA 2.5, SDAID can be invoked in a multi-processor environment (up to 10 CPUs). All in-bound events are processed in parallel mode, that is none of the CPUs are stopped. The exception is the small period of time when records are written to the SDAID buffer.  
Prior to VSE/ESA 2.5, all active processors but one must be stopped prior to invoking SDAID: **SYSDEF TD,STOP=ALL**.
2. SDAID uses the SDAREA as work space. (Defined via **SYS SDSIZE=nnK IPL** command). Some vendor products also use the SDAREA, thus preventing SDAID from being initialized. The default is 64K. Minimum is 48K for two trace statements and no buffer.
3. If output tape fills up, or has intervention required (or an I/O error), SDAID places the system in a wait state with the following PSW:  
000A0000 00EEEEEE. In this case:  
The external interrupt key will be an option under the hardware console. It is probably a good idea to look this up prior to invoking an SDAID trace.

### VTAM SNA:

See the following for information on the VTAM Generalized Trace Facility (GTF):

“VTAM Operations” (SC31-6495-01) under “2.1.87 *MODIFY TRACE Command*”

“VTAM Resource Definition Reference” (SC31-6498-01) under “4.4.111 *TRACE for Buffer, I/O, NCP Line, SIT, or SMS Traces*”

# Tracing Hardware Errors

## CP TRACE:

If:

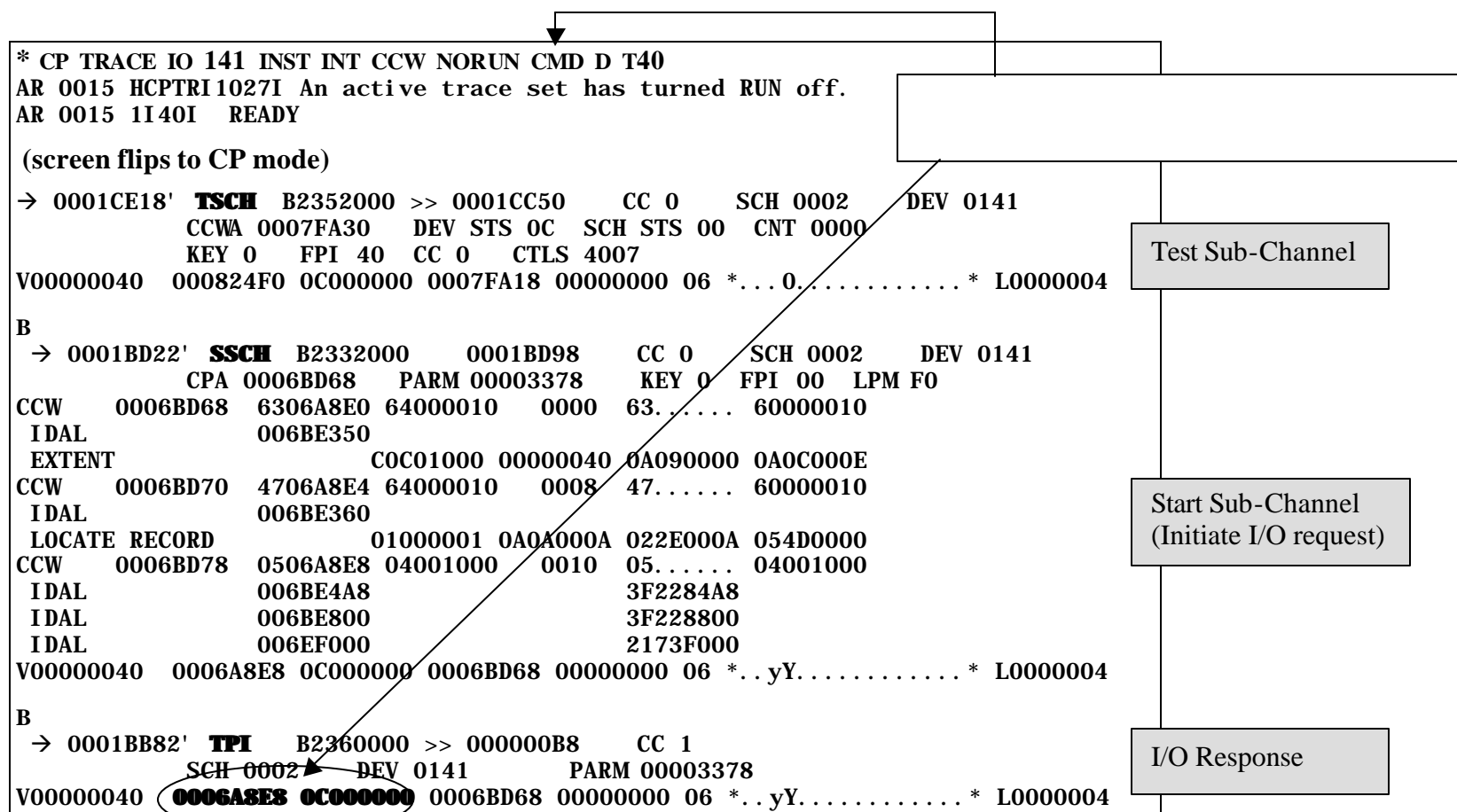
- Running under VM/ESA (zVM)
- Unable to run SDAID
- Amount of trace data is small

1. Trace output goes to console.
2. Not as flexible as SDAID.
3. Translates CCW opcodes into clear text.
4. Specify “NORUN” to avoid lock-out.
5. Initiate trace using:  
\* **CP TRACE IO 141 INST INT CCW NORUN CMD D T40**
6. Terminate trace using:  
\* **CP TRACE END ALL**
7. See example under Speaker Notes.

# Tracing Hardware Errors

## Speaker Notes:

1. The trace output goes straight to the VSE/ESA virtual machine console, so be sure to spool your console before you begin.
2. CP Trace does not have an option to display the data transmitted by / retrieved by CCW commands. Thus, the sense data is not visible in the trace, and must be used from the console message.
4. It is easy to lock your VSE machine out, so specify "NORUN" or be prepared to monitor closely.
6. If the trace has locked up your console, and you cannot get into VSE, try "#CP TRACE END ALL".



## Tracing Hardware Errors

### TRSOURCE / TRSAVE:

1. Can be run concurrently with an SDAID trace.
2. **P**rivilege class “C” CP commands
3. Determine the real device address for the virtual device to be traced.
4. Define the trace sets and traces using **TRSOURCE ID** and **TRSAVE ID** commands.
5. Set up event to be traced.
6. Start trace using **TRSOURCE ENABLE**
7. Reproduce the problem.
8. End the trace using the **TRSOURCE DISABLE** command
9. View the trace data using the **TRACERED** command.

### I/O Tracing for CMS/VSAM

1. VSAM issues its I/Os via SVC0.
2. DMSXCP then executes a Diagnose x'A8'.
3. To trace CMS/VSAM I/O, use: **TRACE DIAG** command

# Tracing Hardware Errors

## Speaker Notes:

Sometimes, if you are running VSE/ESA as a guest under VM/ESA, it is meaningful to use **TRSOURCE** and **TRSAVE** to also trace the VM I/O to the real device. See “*CP Command and Utility Reference*” (SC24-5519-04) The TRSOURCE trace can be run concurrently with an SDAID trace.

3. Determine the real device address for the virtual device defined to the VSE machine:

```
* CP Q V 140
AR 0015 DASD 0140 3390 CC3U3B R/W    1112 CYL ON DASD 7A0E SUBCHANNEL = 0001
AR 0015 1140I  READY
```

4. Define the trace sets and traces to be run using the **TRSOURCE** CP command. This is a privilege class “C” command, so cannot be issued from the VSE/ESA virtual machine or from a normal “G” class user. Define where the trace output should be saved using the **TRSAVE** CP command.

```
TRSOURCE ID IOTRACE TYPE IO DEV 7A0E
TRSAVE ID IOTRACE DASD TO OLSEN NAME VSETEST2 SIZE 8 KEEP 1
TRSOURCE ENABLE IOTRACE
.
.   (Failing event)
.
TRSOURCE DISABLE IOTRACE
```

9. View the trace data in the CP System Trace File (TRFiles) using the **TRACED** command. See “*VM/ESA Dump Viewing Facility*” (SC24-5530-04).

## I/O Tracing for CMS/VSAM

Some have wondered why the CP TRACE I/O doesn't show CMS/VSAM I/O activity. VSAM (or CMSBAM for tape ) issues its I/Os via SVC0. DMSXCP then executes a Diagnose x'A8'. So a real I/O is never executed within the virtual machine.

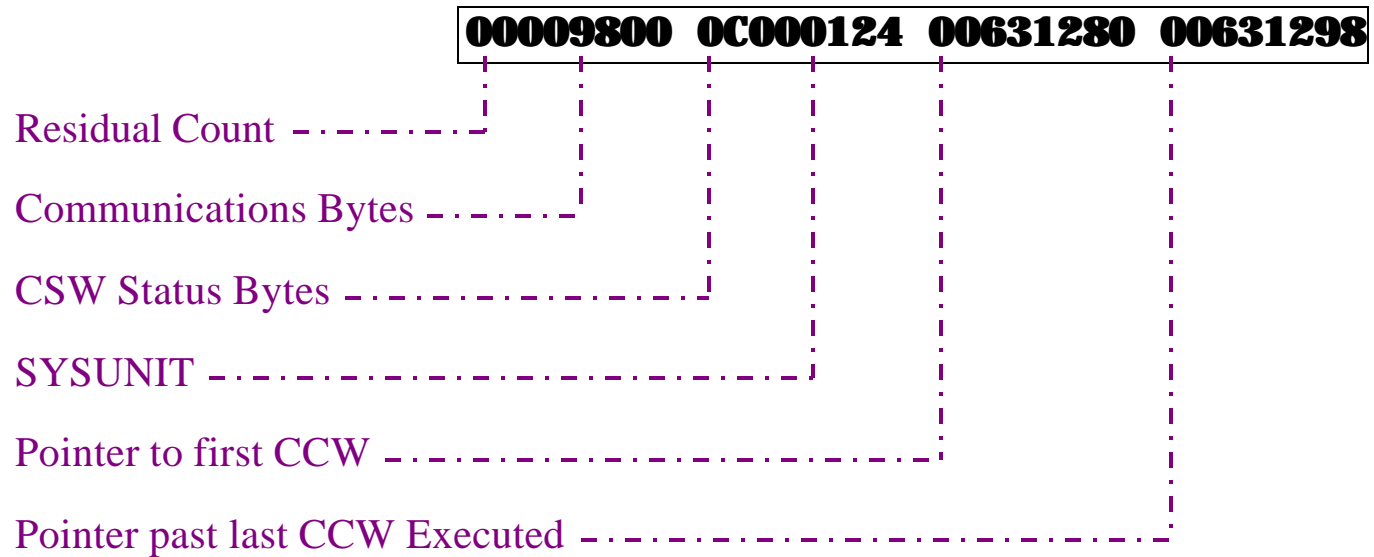
To setup an I/O trace within a virtual machine, you should issue:

```
TRACE DIAG A8 PSWA xxxx.yyy CMD D G
where the PSWA range covers the CMSDOS (or CMSVSAM) segment.
```

# Tracing Hardware Errors

## CCB: (Command Control Block)

- Used by application to request I/O processing from VSE



# Tracing Hardware Errors

## Speaker Notes:

**CCB:** (Command Control Block) used by application to request I/O processing from VSE

**+x'00' Residual Count (two bytes)**

**+x'02' Communications Byte 1**

- x'80' Traffic Bit (Set at Channel End or Device End)  
(Set by VSE)
- x'40' End of File (Set by VSE)
- x'20' Unrecoverable I/O Error (Set by VSE)
- x'10' Accept Unrecoverable Errors (Set by application)
- x'08' Return Data Checks (Set by application)
- x'04' Post at Device End (Set by application)
- x'02' Return Data Check / Read Check (Set by application)
- x'01' User Error Routine (Set by application)

**+x'03' Communications Byte 2**

- x'80' Data Check in count area (Set by VSE)
- x'40' Track Overrun (Set by VSE)
- x'20' End of Cylinder (Set by VSE)
- x'10' Data Check (Set by VSE)
- x'08' No Record Found (Set by VSE)
- x'04' Retry No-record-found (Set by application)
- x'02' Verify Error (Set by VSE)
- x'01' Command Chain (retry) (Set by VSE)

**+x'04' CSW Status Bytes** (Two bytes, see CSW) (Set by VSE)

**+x'06' LUB Class:**

- x'80' EXCP REAL
- x'40' BTAM I/O Request
- x'20' Copied CCB
- x'10' 31-Bit CCW Address
- x'08' Physical Addressing
- x'04' Control Block is an Iorb
- x'01' Programmer Logical Unit

**+x'07 Symbolic Unit** (SYSUNIT) (Set by application)

- If x'01xx': Programmer unit (e.g. x'0124' = SYS036)
- If x'00xx':
  - 0000 = SYSRDR
  - 0001 = SYSIPT
  - 0002 = SYSPCH
  - 0003 = SYSLST
  - 0004 = SYSLOG
  - 0005 = SYSLNK
  - 0006 = SYSRES
  - (and so forth)

**+x'08' Pointer to first CCW** (Set by application)

**+x'0C' Pointer past last CCW executed** (Set by VSE)



# Tracing Hardware Errors

## CCW: (Channel Command Word)

- Specify an I/O command

### Format0 (original) CCW:

**63686940 40000010**

CCW Command Code

Data Address (3 bytes)

Flags

Data Length

### Format1 CCW:

**63400010 00686940**

CCW Command Code

Flags

Data Length

Data Address (4 bytes)

# Tracing Hardware Errors

## Speaker Notes:

### Format0 (original) CCW:

+x'00' CCW Command Code  
+x'01' Data Address (three bytes)  
+x'04' Flags  
x'80' Use address portion of next CCW  
(used with Write Count-Key-Data)  
x'40' Chain to next sequential CCW  
x'20' Suppress Incorrect Length Indication  
x'10' Skip (suppress transfer of data to main storage)  
x'08' Cause Channel Program Controlled Interruption (PCI)  
x'04' Address refers to an Indirect Data Address Word  
+x'05' Unused  
+x'06' Byte Count (two bytes)

### CCW Op-codes commonly-used for DASD:

#### Count-Key-Data (CKD):

07 Seek  
1B Seek Head  
23 Set Sector  
31 Search ID Equal  
08 TIC  
06 Read Data  
05 Write Data  
. 86 Multi-track read

#### Extended-Count-Key-Data (ECKD):

63 Define Extent  
47 Locate Record  
12 Read Count  
06 Read Data  
05 Write Data  
. 86 Multi-track read

### Format1 CCW:

+x'00' CCW Command Code  
+x'01' Flags (see above)  
+x'02' Byte Count (two bytes)  
+x'04' Data Address (four bytes)

### CCW Op-codes commonly used for printers:

09 Space one line after printing  
0B Space one line immediately  
63 Load FCB  
89 Skip to channel one after printing  
8B Skip to channel one immediately

### CCW Op-codes commonly used for tape:

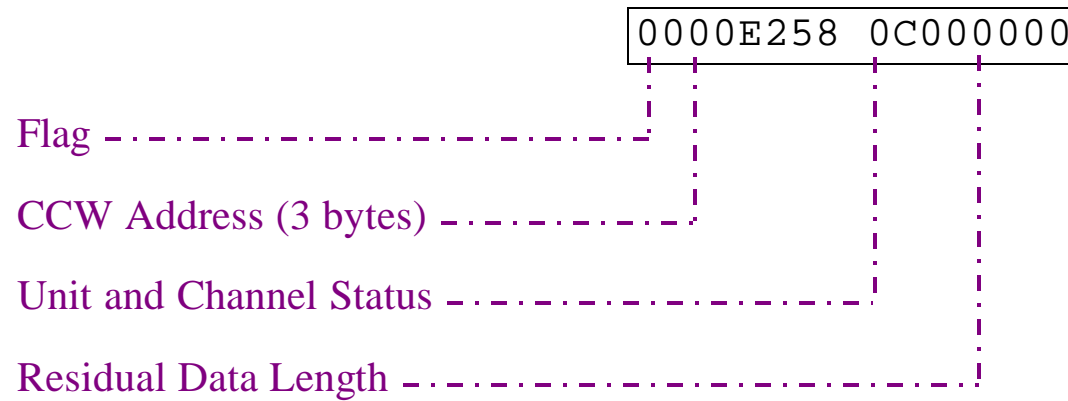
01 Write  
2F Backspace file  
02 Read forward  
3F Forward space file  
07 Rewind  
9F Lead display  
0F Rewind and Unload  
DB Modeset

Sense CCW: 04

# Tracing Hardware Errors

## CSW: (Channel Status Word)

- Returned by hardware to VSE/ESA when I/O complete.



# Tracing Hardware Errors

## Speakers Notes:

### CSW (Channel Status Word):

+x'00'	Multi-use byte		
x'---- . . . .'	Protection Key		
x'08'	Unused		
x'04'	Logout pending		
x' . . . . . --'	Condition code from Start I/O command		
+x'01'	CCW Address (three bytes, real address of last CCW executed)		
+x'04'	Unit Status	+x'05'	Channel Status
<b>x'0C00'</b>	<b>indicates normal completion</b>	x'80'	Program Controlled Interrupt
	(Channel End + Device End)	x'40'	Incorrect Length
<b>x'0D00'</b>	<b>indicates an exceptional condition</b>	x'20'	Program Check
	(CE + DE + Unit Exception)	x'10'	Protection Check
<b>x'0E00'</b>	<b>indicates an error condition</b>	x'08'	Channel Data Check
	(CE + DE + Unit Check)	x'04'	Channel Control Check
x'80'	Attention interrupt	x'02'	Interface Control Check
x'40'	Status Modifier	x'01'	Chaining Check
x'20'	Control Unit End		
x'10'	Busy	+x'06'	Residual Bye count (Difference between bytes requested and number read)
x'08'	Channel End		
x'04'	Device End		
x'02'	Unit Check		
x'01'	Unit Exception		

# Tracing Hardware Errors

## SDAID trace output:

```

START OF SDAID TRACE
SSCH-1 0 BG UNIT=0181
TOD = 2000.054 14.24.42.922
CCB= 0005E7F8 00001000 00002009 0005E8D0 00000000
CCW= 00067460 C30638E5 60000001
CCW= 00067468 0805E8D0
CCW= 0005E8D0 0705E8D0

I/O 00 BG UNIT=0181 CSW =1005E8D8 08000001
TOD = 2000.054 14.24.42.922
CCB= 0005E7F8 00001000 00002009 0005E8D0 00000000
CCW= 0005E8D0 0705E8D0 20000001 **** NO DATA TRANSFERED ****
----IRB-----
0001D320 10404045 0005E8D8 08000001 00400000 00000000 00000000 00000000 00000000
0001D340 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

I/O 00 AR UNIT=0181 CSW =00000000 04000000
TOD = 2000.054 14.24.44.062
CCB CAN NOT BE FOUND
----IRB-----
0001D320 10400003 00000000 04000000 00400000 00000000 00000000 00000000 00000000
0001D340 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

SSCH-1 0 BG UNIT=0181
TOD = 2000.054 14.24.44.402
CCB= 000275D0 00001400 0000081D 000274F8 20027580
CCW= 000274F8 03027F38 60000001
CCW= 00027500 04027F38 20000020
---CCW DATA---
00027F38 00000000 00000000 00000000 00000000 00000000
00027F40 00000000 00000000 00000000 00000000 00000000
.
.
.
  
```

Start Sub-channel  
 Software request out to hardware

Hardware response

Channel End (CE)

Device End (DE)

\* . . . . YQ . . . . . \*  
 \* . . . . . \*

\* . . . . . \*  
 \* . . . . . \*

\* . . . . . \*  
 \* . . . . . \*

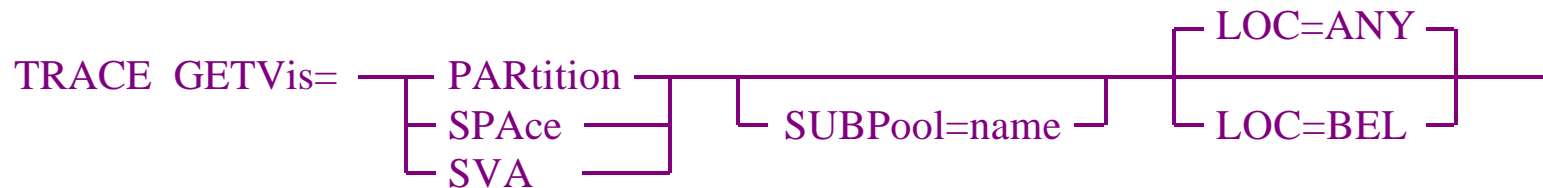


# GETVIS Shortage

## SDAID GETVIS Trace:

See “[VSE/ESA 2.6 Release Guide](#)” / “[8.2 SDAID Enhancements](#)”

```
// EXEC SDAID
OUTDEV P=02E
TRACE GETVIS=PARTITION LOC=ANY AREA=F2 -
      ADDR=0: * OUTPUT=(TOD, GREG) OPTION=NOJCL
/*
/ &
```



Enhancements (over SVC x'3D' / x'3E' trace):

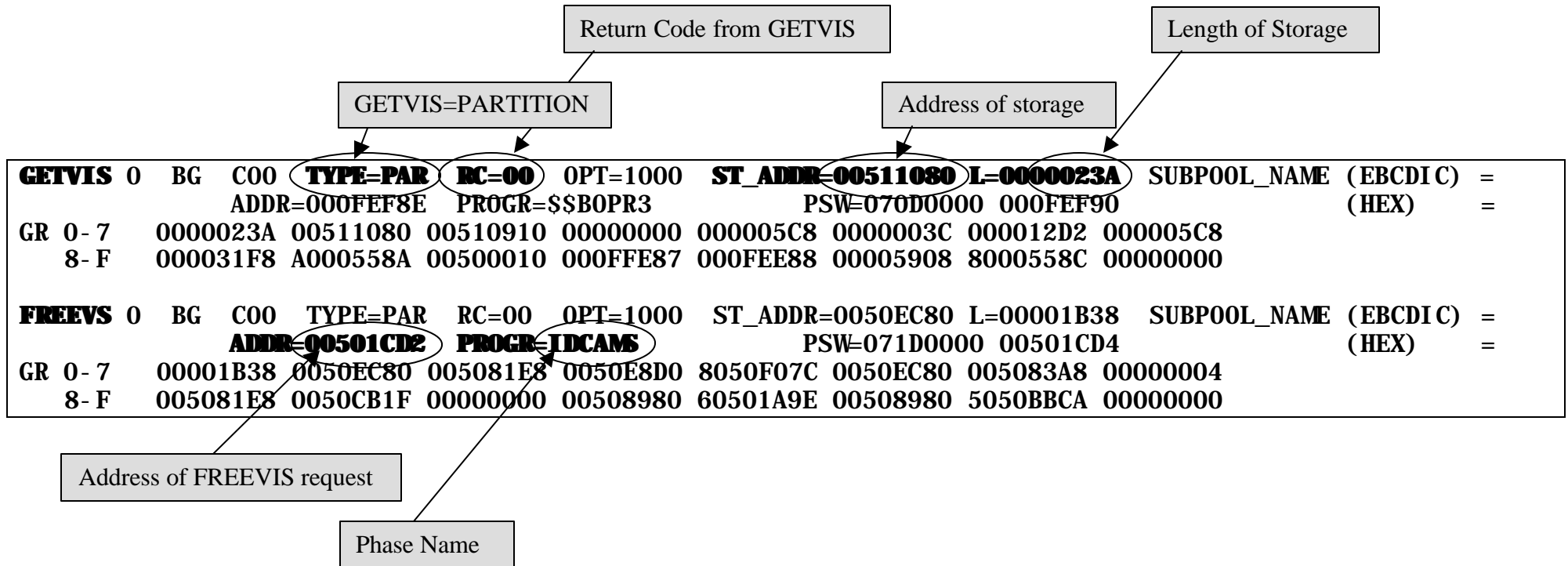
- Shows return code from GETVIS or FREEVIS
- Shows address of storage retrieved by GETVIS

Requires post-processing to identify storage which was requested but not released.

# GETVIS Shortage

## Speaker Notes:

- Implemented in VSE/ESA release 2.6. Retrofitted to 2.5 via APAR DY45830
- Do not use "SUBPOOL=ALL". ("ALL" will be interpreted as name of subpool)
- AREA=<syslogic>" by itself does not trace SVA activity on behalf of this partition. Add "ADDR=0:\*".
- AREA=<syslogid>" must be coded prior to "ADDR=0:\*"





# VSAM File Access Conflicts

## Symptom:

*“Unexplained errors running VSAM operations.*

*File being accessed is SHR(2), and not shared between multiple VSE systems.”*

```
F4 004 4228I FILE KSDS          OPEN  ERROR X' A8' (168) CAT=UCAT  
(OCSHR- - 5) FILE ALREADY OPEN IN ANOTHER PARTITION
```

```
F4 004 4228I FILE KSDS          OPEN  ERROR X' FF' (255) CAT=UCAT  
(OCSHR- - 8) LOCK RETURN CODE X' 0C'
```

```
DELETE (CICS. CSD) CLUSTER PURGE  
IDC3028I DATA SET IN USE  
IDC3009I ** VSAM CATALOG RETURN CODE IS 184 - REASON CODE IS IGGOCLCX-4  
IDC0551I **ENTRY CICS. CSD NOT DELETED  
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
```

## Resolution:

1. LOCK TRACE=F4,V\* (replace “F4” with failing partition)
2. Re-run failing job
3. LOCK TRACE=OFF
4. Locate failing LOCKTAB and OWNER trace entry

# VSAM File Access Conflicts

## Speaker Notes:

File is open in another partition, but you don't know which one. Use the LOCK TRACE command to identify the partition that has the file open (See "Hints and Tips for VSE", p. 33) If you receive a msg4228I on the console:

1. Locktab entries and Owner elements are matched pairs.
2. The Locktab and Owner entry is immediately prior to the msg4228I in the console log identifies the failing LOCK request.
3. Having located the failing locktab entry, move to the Owner element immediately following.  
+5 (counting from zero) contains the task id of the owner of the file in question

<u>Task id</u>	<u>Partition id</u>	<u>Storage Key</u>
21	BG	1
22	F1	C
23	F2	B
24	F3	A
25	F4	9
26	F5	8
27	F6	7
28	F7	6
29	F8	5
2A	F9	4
2B	FA	3
2C	FB	2
2D-FF	Dynamic Partitions	D

Use STATUS command to identify a dynamic partition (or VSE/ESA sub-task) belonging to a task id higher than x'2C'.

# VSAM File Access Conflicts

## Sample Console Output:

```
LOCK TRACE=F4
AR 0015 1I40I  READY
... (skip several pages of trace data)
F4 0025 LOCKTAB ENTRY
V02CCCAE0  02CCCB10 00000000 E5C3E3E2 F2F2F000 * öô      VCTS220 *  R0096DAE0
V02CCCAF0  00010001 04C00000 02CCCAC0 02CCCB20 * î    {  ö- { öô *  R0096DAF0
F4 0025 OWNER ELEMENT
V02CCCB10  00000000 00210001 00000000 00000000 *                *  R0096DB10
F4 0025 LOCKTAB ENTRY
V02CCCAE0  02CCCB10 00000000 E5C3E3E2 F2F2F000 * öô      VCTS220 *  R0096DAE0
V02CCCAF0  00560001 04C00000 02CCCAC0 02CCCB20 * î    {  ö- { öô *  R0096DAF0
F4 0025 OWNER ELEMENT
V02CCCB10  00000000 00210001 00000000 00000000 *                *  R0096DB10
F4 0004 4228I FILE KSDS      OPEN  ERROR X' A8' (168) CAT=IJSYSCT
(OPNAB-15) DATASET ALREADY BEING LOADED BY ANOTHER ACB
```

CI# of failing file in  
VSE/VSAM Catalog

Taskid owning lock

Volume where  
catalog resides

# VSAM File Access Conflicts

## Speaker Notes:

In cases where no console error message is issued (e.g. IDCAMS DELETE “*Data Set in Use*”), the failing trace entry will be more difficult to find. For an IDCAMS DELETE, it will be towards the end of all the trace entries for this job.

1. Scan the trace entries ignoring all locktab entries which do not contain "V" plus the volume name of the user catalog where the file in question resides.
2. Ignore all locktab entries in the following format: "**V**"<volser>nnnnnnmmmm, where nnnnnn = '000000' or '000001'.
3. We are looking for a locktab entry in the following format: "**V**"<volser>cccccc**0001**" :. "cccccc" will be the control interval number of the file in question within the user catalog. In our case, the user catalog resides on volume CTS220, and the file is ci# x'000056': e.g. **x'E5C3E3E2 F2F2F000 00560001'** (VCTS220..1..). "0001" may also be "0000".
4. In some cases, there will be two locktab entries following each other, that match this format. This would be for a KSDS file, and the first one describes the data component, and the second, the index.
5. Terminate the lock trace using "**LOCK TRACE=END**"

# VSAM File Access Conflicts

## Identify filename under which file is opened:

- Use the LOCATE command to find the AMBL:

```
LOCATE BG,'TEST.KSDS
AR 0015  MATCH FOUND AT 004044B0
V004044B0 E3C5E2E3 4BD2E2C4 E24BC3D3 E4E2E3C5 16 *TEST. KSDS. CLUSTE* R00D7F4B0
V004044C0 D9404040 40404040 40404040 40404040 16 *R * R00D7F4C0
AR+0015
15 E
```

- Backup x'98'. The VSAM AMBL starts with '110000C4'. AMBL+x'14 points at the ACB.

```
SHOW BG,404418.40
AR 0015  DATA  FOUND AT 00404400
V00404410 ..... 110000C4 004045E0 16 *..... \... D. .\* R00D7F41
V00404420 00404780 00402D18 00404418 00400494 16 *. ... .. .mf R00D7F42
V00404430 02A00100 00003A00 9A000422 00020003 16 *..... * R00D7F43
:
V004044B0 E3C5E2E3 4BD2E2C4 E24BC3D3 E4E2E3C5 16 *TEST. KSDS. CLUSTE* R00D7F4B0
V004044C0 D9404040 40404040 40404040 40404040 16 *R * R00D7F4C0

SHOW BG,400494.40
V00400490 ..... A040004C 00404418 001B3FF8 16 *..... <. .... 8* R00D8F490
V004004A0 00000000 00009A00 28110400 00000000 16 *..... * R00D8F4A0
V004004B0 D2E2C4E2 40404040 00000000 00000000 16 *KSDS * R00D8F4B0
V004004C0 0040A000 00000000 00000000 00080000 16 *..... * R00D8F4C0
V004004D0 00000000 16 *..... * R00D8F4D0
```

# VSAM File Access Conflicts

## Following errors will result in OPEN return code x'A8':

1. Attempt to OPEN a SHR(2) file for output while the same file is already open for output in another partition or same partition. Extended message will tell customer if file is open in this partition or another partition. Remember that OPENS over a path (AIX or base) count,
2. KSDS files can be opened in one of three modes: Keyed, Addressed, or CI-mode. Once a SHR(4) file is opened in one of these modes, all other OPENS (even from a different partition) must be in the same mode. This often happens when sharing files between CICS partitions, particularly with the IESCNTL file. IESCNTL must be specified as SHR(4) if it is to be shared between CICS partitions.
3. If the file is shared between multiple VSE/ESA systems, ensure that the CPUID has not changed. The external locks include the CPUID in the key, and if the CPUID is changed, old locks under the old CPUID are not cleared. If VSE systems are run second level under VM, changing the CPUID is often done using the "CP SET CPUID" which will, in turn, drive a different ASI proc. Also, powering off a secondary system without first closing all VSAM files, may also leave ghost lock entries in the lock file, which are not freed until the next IPL. Use the **AR UNLOCK** command to release lock entries for the bogus CPUIDs.
4. If files are shared between multiple releases of VSE/ESA, ensure that DOSRES or SYSWK1 are not marked as ",SHR" on both systems. Even though each system has their own version of DOSRES or SYSWK1, or only link to the appropriate volume, all locks are done using only the volume of the catalog and the CI number within the catalog. Thus, VSAM may pick up a lock from one release / catalog, and find that it matches the CI from the other release / catalog, and reject the OPEN.
5. When a file is being initially loaded (prior to first CLOSE), the SHAREOPTIONS for the file are temporarily reset to '1'. This means, even if the SHAREOPTIONS are '2' or '4', another partition will get rc\_x'A8' attempting to open the file, even for input.
6. Close of a file under CICS may not always be successful. This is the case whether the close is issued via CEMT command, or from one of a number of available vendor products which issue a request from batch to CICS to close an on-line file so it can be opened for output in batch. CICS will not close the file if there is an outstanding request against the file. This includes an update request that has not been either returned or released to File Control.
7. Consider the following case:
  - File defined with SHR(2)
  - The file is defined multiple times to CICS (or base and path are both defined for the same file)
  - The file definitions are made with "Dataset Name Sharing" ("DSNSHARING" under CICS TS CEDA, "BASE=" under CICS/VSE)
  - At least one file definition is made with **OPERATIONS** of **UPDATE** ("SERVREQ OUTPUT" under CICS/VSE)CICS will adjust all file accesses to output. Then if file originally defined as "UPDATE" is closed, the remaining FCT entries still maintain the output lock for the file, even if they were originally defined as "BROWSE". This will prevent opening the file in another partition.

## Loop during CICS Initialization

### Symptom:

CICS crashed. When we-ipl-ed then attempted to bring it back up, it hangs during initialization.

### Resolution:

- 1) Use **STATUS F2** to check CICS partition
- 2) Use **SHOW** to display task save area.
  - a) Display address in current PSW.
  - b) Locate active module.
- 3) If “IKQ...”, this is VSAM, and probable corrupted file index.
- 4) Determine which file is currently being accessed.
  - Reg11: AMBL (Extract 44-character fileid)
  - Reg12: RPL
  - Reg13: PLH
  - ACB (Extract 8-character filename).
- 5) Use **IDCAMS REPRO** to test file. If it also loops, re-build this file.

# Loop during CICS Initialization

## Speaker Notes:

1. Check status of CICS partition using **STATUS** command. If CICS/VSE, check status of partition main task. If CICS TS, check status of Quasi-Reentrant (second DFHEVID1) sub-task. If task shows “Ready to Run”, it may be in a loop.
2. Display task save area using **SHOW** command. Display address in current PSW. Identify failing module.
  - Scan code around where the PSW is pointing for a “Branch” instruction. Extract the base register and display around that address looking for an eyecatcher. –or–
  - Scan backwards looking for an eyecatcher.
3. If active module (phase) eyecatcher begins with “IKQ”, this is VSAM, and we are probably suffering a corrupted file index. If CICS (or the VSE/ESA system) goes down hard, without allowing VSE/VSAM Automatic Close to flush the active buffers, partial updates may be left on disk, which may be incompatible with the previous data.
4. Determine what file is currently being accessed. Check contents of Reg10 – Reg13. At least one of them will be pointing at one of the following VSAM control blocks:
  - **AMBL** begins with “110000C4”. ACB pointer is at displacement x’14’. The 44-character fileid is visible at x’78’.
  - **ACB** begins with “A040004C”. AMBL pointer is at displacement x’08’. The 8 character filename used to open the file is at x’1C’.
  - **PLH** (Place Holder) has “55FF” at displacement x’114’, followed by a pointer to the ACB at x’134’. Two pointers to the current RPL at are x’C8’ and x’CC’.
  - **RPL** begins with “00100034”. ACB pointer is at displacement x’18’.
5. If you are successful in identifying a failing file, take it off-line, and attempt to bring up CICS without this file. Print the entire file, or copy it to tape using **IDCAMS REPRO**. Often this will also show a loop. At this point you will have to re-build the file: Often, if only the index is corrupted, you can recover the data component of a file by:
  - **REPRO** the data component of the file to tape
  - Sort the output tape (eliminate duplicates)
  - Delete and re-define the file
  - Re-load the file from the sorted output tape.



# Loop during CICS Initialization

## 1. Check status of CICS partition

CICS/VSE:

```
STATUS F2
AR 0015 S3E- F2 DFHLOADR 82 WAITING FOR I/O, OR ECB POSTING
AR 0015     SCB=0004088C PCB=00041280 TCB=0025907C TIB=00259000 SAV=008F90E0
AR 0015 S3F- F2 DFHSKP   82 WAITING ON TIMER INTERRUPT
AR 0015     SCB=0004088C PCB=00041280 TCB=0025937C TIB=00259300 SAV=004D1D34
AR 0015 S40- F2 DTSCOPCM 82 WAITING ON TIMER INTERRUPT
AR 0015     SCB=0004088C PCB=00041280 TCB=0025967C TIB=00259600 SAV=00903028
AR 0015 S41- F2 DTSNTFY  82 WAITING ON TIMER INTERRUPT
AR 0015     SCB=0004088C PCB=00041280 TCB=0025997C TIB=00259900 SAV=0091DA90
AR 0015 S42- F2 DTSCHIGH 82 WAITING ON TIMER INTERRUPT
AR 0015     SCB=0004088C PCB=00041280 TCB=00259C7C TIB=00259C00 SAV=0094F8A0
AR 0015 M23 F2 CICSICCF 83 READY TO RUN
AR 0015     SCB=0004088C PCB=00041280 TCB=0004AA80 TIB=00043680 SAV=00400000
AR 0015 1I40I  READY
```

CICS TS:

```
STATUS F2
AR 0015 S40- F2 EVA10MST 82 WAITING FOR I/O, OR ECB POSTING
AR 0015     SCB=0004980C PCB=0004A200 TCB=002C461C TIB=002C45A0 SAV=00405260
AR 0015 S41- F2 DFHEVID2 82 WAITING FOR I/O, OR ECB POSTING
AR 0015     SCB=0004980C PCB=0004A200 TCB=002C48EC TIB=002C4870 SAV=00292400
AR 0015 S42- F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015     SCB=0004980C PCB=0004A200 TCB=002C4BBC TIB=002C4B40 SAV=00292480
AR 0015 S43- F2 DFHEVID1 83 READY TO RUN
AR 0015     SCB=0004980C PCB=0004A200 TCB=002E607C TIB=002E6000 SAV=00292500
AR 0015 S44- F2 DFHSKTSK 82 WAITING FOR I/O, OR ECB POSTING
.
.
.
AR 0015 M23 F2 CICSICCF 82 WAITING FOR I/O, OR ECB POSTING
AR 0015     SCB=0004980C PCB=0004A200 TCB=00054E40 TIB=0004C600 SAV=00400000
```

## Loop during CICS Initialization

### 2. Display task save area

```

SHOW F2,400000.60
AR 0015 DATA FOUND AT 00400000
V00400000 C4C6C8E2 C9D74040 07ED2000 801ADE18 B6 *DFHSIP ..... * R01310000
V00400010 009128E0 0090C990 00912910 001AD578 CB6 *.j.\..I..j....N.* R01310010
V00400020 012E4798 801AEC26 001AF7E9 801AF30E B6 *...q.....7Z..3.* R01310020
V00400030 009128C0 00000000 012E970C 00912840 B6 *.j.{.....p..j.* R01310030
V00400040 004CE8BC 801ADE1C 0090CA98 012F1398 B6 *.<Y.....q...q* R01310040
V00400050 0000B3D9 A4C8980D 40404040 40404040 B6 *..RuHq.* R01310050
    
```

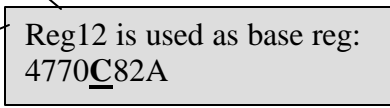
2. ... Display address in current PSW



```

SHOW 1ADE10.40
V001ADE10 12114770 C82A07F6 5810D050 58101070 06 *...H..6..}&... * R00624E10
V001ADE20 911010A6 07E6185B 58B0D050 4510C8E0 06 *.j..w.W.S..}&..H\ * R00624E20
V001ADE30 5B5BC2C3 E5E2F0F3 C9D2D8E5 C4E4D4D7 06 *$$BCVS03IKQVDUMP * R00624E30
V001ADE40 FFFFFFF67 00000000 001AD57C FFFFFFFF 06 *.....N@.... * R00624E40
AR 0015 1I40I READY
    
```

2. ... Identify failing module.



```

SHOW F2,001AD578.60
V001AD570 ..... 47F0C018 C9D2D8C9 06 *.....0{.IKQI * R00624570
V001AD580 D6C44040 F3F5C340 C4E8F4F4 F7F7F800 06 *OD 35C DY44778.* R00624580
V001AD590 4560C8A0 5860D108 58106030 12114780 06 *. -H.. -J... -..... * R00624590
V001AD5A0 C0D49110 10024710 C0CA1B55 BF571009 06 *{Mj.....{..... * R006245A0
V001AD5B0 951B5000 4770C044 92075000 58501020 06 *.n.&...{.k.&..&.. * R006245B0
V001AD5C0 4A55003A 4B50C900 ..... 06 *.c....&I.* R006245C0
    
```

## Loop during CICS Initialization

### 4. Determine which file is currently being accessed

- Register 11 often points to the AMBL:

Eyecatcher

Pointer to ACB  
AMBL+x'14'

```

SHOW F2,0090C818.100
V0090C810 ..... 110000C4 0090C990 B6 * ..... D..I.* R01413810
V0090C820 0090CB30 012E4798 0090C818 008F1BF4 B6 * ..... q..H...4* R01413820
V0090C830 02A80380 00005600 8A0005D4 00040009 B6 *.y.....M...* R01413830
V0090C840 00000000 00008000 00008000 FFFFFFFF B6 * ..... * R01413840
V0090C850 FFFFFFFF 00800000 00000000 012F1798 B6 * ..... q* R01413850
V0090C860 00000000 0090E800 00000000 00000000 B6 * ..... * R01413860
V0090C870 00000000 00000000 00000000 00000000 B6 * ..... * R01413870
V0090C880 00008000 00000000 00003F00 00000000 B6 * ..... * R01413880
V0090C890 01000000 00000000 00000000 00000000 B6 * ..... * R01413890
V0090C8A0 001BFBC0 00000000 00000000 00000000 B6 * ..... * R014138A0
V0090C8B0 C3C9C3E2 4BD9E2C4 40404040 40404040 B6 * *CICS.RSD* * R014138B0
V0090C8C0 40404040 40404040 40404040 40404040 B6 * ..... * R014138C0
V0090C8D0 40404040 40404040 40404040 00000000 B6 * ..... * R014138D0
    
```

Fileid  
AMBL+x'98'

- Register 12 often points to the RPL:

Eyecatcher  
CICS TS: "0011003C"

```

SHOW F2,004CE8BC.40
V004CE8B0 ..... 00100034 B6 * ..... * R014368B0
V004CE8C0 00000125 004CE881 004E63F4 000000A6 B6 * ..... <Ya.+.4...w* R014368C0
V004CE8D0 00000004 008F1BF4 011C0006 0800FFFF B6 * ..... 4..... * R014368D0
V004CE8E0 00000000 00000000 00000000 00000000 B6 * ..... * R014368E0
    
```

Pointer to ACB  
RPL+x'18'

## Loop during CICS Initialization

- Register 13 often points to the **PLH** (Placeholder):

```

SHOW F2,012E4798.140
V012E4790 ..... 00000000 005113C8 B6 * ..... H* R01416790
V012E47A0 00000000 00000066 012E4904 012E96F0 B6 * ..... o0* R014167A0
V012E47B0 00901300 00000000 008F1BF4 00901E80 B6 * ..... 4. . . * R014167B0
.
.
V012E4810 801CEA ..... 00 00000000 B6 * ..... 6f. .... * R01416810
V012E4820 801CD ..... 01 00000013 B6 * ..... * R01416820
V012E4830 02115210 801CC920 00D34098 00000000 B6 * ..... I.. N q. . . * R01416830
V012E4840 801DF53A 001CCF70 001CC7A0 001DF400 B6 * .. 5. .... G. . . 4. * R01416840
V012E4850 00000000 00000000 00000000 0 ..... * R01416850
V012E4860 004CE8BC 004CE8BC 00008000 F ..... * R01416860
V012E4870 FFFFFFFF 801AEB82 00000000 0 ..... * R01416870
V012E4880 00000000 00000000 00000000 00000000 B6 * ..... * R01416880
V012E4890 00000000 00000000 00000001 00460015 B6 * ..... * R01416890
V012E48A0 012F1398 012E96F0 012F138C 55FF0023 B6 * .. q. . o0. .... * R014168A0
V012E48B0 000CA900 012E4A28 00000000 00000000 B6 * ..... * R014168B0
V012E48C0 00000000 00000000 00000000 008F1BF4 B6 * ..... 4 * R014168C0
V012E48D0 00000000 00000000 01FF0023 00901300 B6 * ..... * R014168D0
V012E48E0 00000000 001AE9C8 00000000 00000000 B6 * ..... ZH. .... * R014168E0
V012E48F0 012F1398 012F1298 004CE8BC 00000000 B6 * .. q. . q. <Y. .... * R014168F0
  
```

Annotations in the code block:

- Box: "Pointers to Active RPL PLH+x'68 and x'6C'" with arrows pointing to 801CEA and 801CD.
- Box: "Eyecatcher PLH+x'114'" with arrows pointing to 004CE8BC and 004CE8BC.
- Box: "Pointer to ACB PLH+x'134'" with an arrow pointing to 008F1BF4.

**ACB:**

```

SHOW F2,008F1BF4.60
V008F1BF0 ..... A040004C 0090C818 001A8450 B6 * ..... d&* R013F6BF0
V008F1C00 00000000 00008A00 28110300 00000000 B6 * ..... * R013F6C00
V008F1C10 C4C6C8D9 E2C44040 00000000 00000000 B6 * ..... * R013F6C10
V008F1C20 012D3A00 004A5FB4 00000000 00080000 B6 * ..... C- . . . . * R013F6C20
V008F1C30 00000000 00000000 01000000 00000000 B6 * ..... * R013F6C30
V008F1C40 00100034 B6 *.... * R013F6C40
  
```

Annotations in the code block:

- Box: "Eyecatcher" with an arrow pointing to A040004C.
- Box: "Pointer to AMBL ACB+x'04'" with an arrow pointing to 0090C818.
- Box: "Filename ACB+x'1C'" with an arrow pointing to DFHRS0.

# Correlate VSE/ESA Dump with COBOL Source

## Symptom:

CICS Partition hang. No response to MSG F2. The partition appears to be looping. Repeatedly issuing the STATUS command, and checking the second DFHEVID1 save area shows PSWs in a limited range of storage.

## Resolution:

1. Use **STATUS F2** to check failing partition
2. Use **SHOW** to display task save area.
  - a) Display address in current PSW.
  - b) Locate active module.
3. Identify name of COBOL program.
4. Locate LE/VSE stub then DFHELII stub.
5. Calculate displacement from DFHELII stub to PSW location.
  - a) Locate this displacement in LNKEDT listing. Identify failing routine.
  - b) Calculate displacement from beginning of failing routine to PSW location.
6. Locate this displacement in COBOL Listing (requires CBL Option LIST).
  - a) Identify failing COBOL statement.

# Correlate VSE/ESA Dump with COBOL Source

## 1. Use STATUS F2 to check CICS partition

### STATUS F2

```
AR 0015 S3E- F2 EVA10MT 83 READY TO RUN
AR 0015      SCB=0004908C PCB=00049A80 TCB=003B907C TIB=003B9000 SAV=00605260
AR 0015 S3F- F2 DFHEVID2 83 READY TO RUN
AR 0015      SCB=0004908C PCB=00049A80 TCB=003B934C TIB=003B92D0 SAV=00391300
AR 0015 S40- F2 DFHEVID1 83 READY TO RUN
AR 0015      SCB=0004908C PCB=00049A80 TCB=003B961C TIB=003B95A0 SAV=00391380
AR 0015 S41- F2 DFHEVID1 83 READY TO RUN
AR 0015      SCB=0004908C PCB=00049A80 TCB=003B98EC TIB=003B9870 SAV=00391400
AR 0015 S42- F2 DFHSKISK 82 WAITING FOR I/O, OR ECB POSTING
AR 0015      SCB=0004908C PCB=00049A80 TCB=003B9BEC TIB=003B9B40 SAV=00391480
AR 0015 M23 F2 DEDCCICS 83 READY TO RUN
AR 0015      SCB=0004908C PCB=00049A80 TCB=000546C0 TIB=0004BE80 SAV=00600000
```

Task Save Area.  
Display this area

## Correlate VSE/ESA Dump with COBOL Source

### 2. Use SHOW to display task save area.

SHOW 391400.60

AR 0015 DATA FOUND AT 00391400

```
V00391400 C4C6C8C5 E5C9C4F1 07ED0000 81B005EE 06 *DFHEVID1....a... * R00256400
V00391410 0165B710901B00144A01B003F4B01B00114C 06 *.....0.... * R00256410
V00391420 0165B4C8D81B00544E00000000F0165B60C0 06 *...Ha..... * R00256420
V00391430 0165B67810000000020165B820301B000584 06 *..... * R00256430
V00391440 0165B190500710B9C6000000007007C00D08 06 *.....@.} * R00256440
V00391450 0000B1FF 004190CF 40404040 40404040 06 *..... * R00256450
AR 0015 1I40I READY
```

2 b/c. Display address in current PSW. Locate active module.

SHOW 1B003D0.60

```
V01B003D0 00002802 33800000 000040C0 00014000 B6 *..... * R003003D0
V01B003E0 07080000 28023302 C0000808 00002802 B6 *..... * R003003E0
V01B003F0 33404040 5820D130 D2032000 A010D203 B6 * ..J.K....K * R003003F0
V01B00400 2008A010 D2032010 A010D207 2018A243 B6 *....K....K.... * R00300400
.
.
.
V01B005C0 58B0C02C 47F0B49A 58B0C02C 47F0B382 B6 *.....0.....0.. * R003005C0
V01B005D0 58B0C02C 47F0B322 58B0C02C 47F0B352 B6 *.....0.....0.. * R003005D0
V01B005E0 58B0C02C 47F0B46A D24F9110 9000D201 B6 *.....0..K....K * R003005E0
V01B005F0 9240A014 47F0B1F4 D2109538 A2039240 B6 *.. ...0.4K..... * R003005F0
```

Register 11 is base reg

- We don't appear to have an eyecatcher around the base register, so we fall back to scrolling backwards.

## Correlate VSE/ESA Dump with COBOL Source

This page  
left  
intentionally  
blank



## Correlate VSE/ESA Dump with COBOL Source

### 3. Identify name of COBOL program, then locate LE/VSE stub and DFHELII stub

As we scroll back, we notice a distinctive pattern. Preceding the beginning of code (pointed to by the base register), we encounter an area with mixed hex and declarative literals (CGT and PGT). Toward the end (high address) of this literal pool, we see the name of the COBOL program.

As we scroll back further, we again see the name of the COBOL program with a compile date/time. x'80' bytes prior to this we see 'DFHYI410', which is the eyecatcher for DFHELII.

DFHELII	<b>C4C6C8E8</b>	<b>C9F4F1F0</b>	<b>58F00014</b>	<b>58F0F0B4</b>	<b>B6</b>	<b>* DFHYI410. 0... 00. *</b>	<b>R00673000</b>
	<b>58F0F00C</b>	<b>58FF000C</b>	<b>07FF0000</b>	<b>00000000</b>	<b>B6</b>	<b>* . 00..... *</b>	<b>R00673010</b>
<b>VO1B00020</b>	<b>47F0F028</b>	<b>00C3C5C5</b>	<b>00000000</b>	<b>00000014</b>	<b>B6</b>	<b>* . 00.. CEE..... *</b>	<b>R00673020</b>
<b>VO1B00030</b>	<b>47F0F001</b>	<b>4ACEAC00</b>	<b>01B000CC</b>	<b>00000000</b>	<b>B6</b>	<b>* . 00. c..... *</b>	<b>R00673030</b>
	<b>000</b>	<b>00000000</b>	<b>90ECD00C</b>	<b>4110F038</b>	<b>B6</b>	<b>* ..... }... 0. *</b>	<b>R00673040</b>
LE/VSE Preface	<b>04C</b>	<b>07FF0000</b>	<b>01B00020</b>	<b>01B00114</b>	<b>B6</b>	<b>* q. 0&lt;..... *</b>	<b>R00673050</b>
<b>VO1B00060</b>	<b>01B009F0</b>	<b>01B00080</b>	<b>01B00020</b>	<b>01B00472</b>	<b>B6</b>	<b>* ... 0..... *</b>	<b>R00673060</b>
<b>VO1B00070</b>	<b>01B01078</b>	<b>01B000E0</b>	<b>00000000</b>	<b>00000007</b>	<b>B6</b>	<b>* ..... \..... *</b>	<b>R00673070</b>
<b>VO1B00080</b>	<b>C3D6C2E3</b>	<b>C5E2E340</b>	<b>F1F9F9F9</b>	<b>FOF3F2F5</b>	<b>B6</b>	<b>* COBTEST 19990325 *</b>	<b>R00673080</b>
<b>VO1B00090</b>	<b>F1F6F1F9</b>	<b>F2F2F0F1</b>	<b>FOF1F0F0</b>	<b>00000000</b>	<b>B6</b>	<b>* 161922010100.... *</b>	<b>R00673090</b>
⋮							
			COBOL program name				
⋮							
<b>VO1B00150</b>	<b>00000001</b>	<b>00000000</b>	<b>0050000D</b>	<b>000200C8</b>	<b>B6</b>	<b>* ..... &amp; .... H*</b>	<b>R00673150</b>
<b>VO1B00160</b>	<b>E28896A4</b>	<b>93844095</b>	<b>96A34099</b>	<b>85838589</b>	<b>B6</b>	<b>* Should not recei*</b>	<b>R00673160</b>
<b>VO1B00170</b>	<b>A58540A3</b>	<b>8889A240</b>	<b>85999996</b>	<b>99404DC9</b>	<b>B6</b>	<b>* ve this error (I*</b>	<b>R00673170</b>
<b>VO1B00180</b>	<b>93939687</b>	<b>898340C5</b>	<b>99999699</b>	<b>5D407E7E</b>	<b>B6</b>	<b>* Ilogic Error) ==*</b>	<b>R00673180</b>
<b>VO1B00190</b>	<b>6ED58194</b>	<b>85409596</b>	<b>A3408995</b>	<b>40C995A3</b>	<b>B6</b>	<b>* &gt;Name not in Int*</b>	<b>R00673190</b>
⋮							
						Literal Pool (CGT & PGT)	

## Correlate VSE/ESA Dump with COBOL Source

V01B00330	A3407E7E	6ED596A3	40C1A4A3	88969989	B6	*t ==>Not Authori*	R00673330
V01B00340	A9858440	7E7E6E06	06E00007	00004000	B6	*zed ==>..\.... *	R00673340
V01B00350	FOFOFOF6	F5404040	C960D640	C5999996	B6	*00065 I-O Erro*	R00673350
V01B00360	99407E7E	6ED2C5E8	FOFOFOFO	FOFOF1D2	B6	*r ==>K*Y0000001K*	R00673360
V01B00370	E2C4E240	404040C3	D6C2E3C5	E2E34040	B6	*SDS <b>COBTEST</b> *	R00673370
V01B00380	40404040	404040E2	E8E2D6E4	E3404000	B6	* SYSOUT *	R00673380
V01B00390	00000000	00013400	00000100	00013800	B6	* .....	R00673390
V01B003A0	00000300	00000000	00000000	00000080	B6	* .....	R006733A0
V01B003B0	00000040	00000000	00000000	00000000	B6	* .....	R006733B0
V01B003C0	00000080	00000000	25C00001	C0000808	B6	* .....{..{*	R006733C0
V01B003D0	33800000	000040C0	00014000	00014000	B6	* .....{.. *	R006733D0
V01B003E0	00000000	28023302	C0000808	00002802	B6	* .....{..... *	R006733E0
V01B003F0	40404040	5820D130	D2032000	A010D203	B6	* .. J. K. .... K *	R006733F0
V01B00400	2008A010	D2032010	A010D207	2018A243	B6	* .... K. .... K .. s. *	R00673400
V01B00410	D2032020	A010920E	2028920F	2030D203	B6	* K. .... k. .. k. .. K *	R00673410
.							
.							
.							
V01B005C0	D203202C	47F0B49A	58B0C02C	47F0B382	B6	* .. {.. 0. .... {.. 0. b*	R006735C0
V01B005D0	D203202C	47F0B322	58B0C02C	47F0B352	B6	* .. {.. 0. .... {.. 0. *	R006735D0
V01B005E0	58B0C02C	47F0B46A	D24F9110	9000D201	B6	* .. {.. 0. . K j. . . K *	R006735E0
V01B005F0	9240A014	47F0B1F4	D2109538	A2039240	B6	* k ... 0. 4K. n. s. k *	R006735F0

Base Register

PSW points here

# Correlate VSE/ESA Dump with COBOL Source

## 5. Calculate displacement from DFHELII stub to PSW location.

```
// EXEC LNKEDT, SIZE=256K
JOB TEST2CMP 03/25/1999 5686-066-06-3
ACTION TAKEN MAP
```

PSW points to: 1B005EE, which is x'5EE' past DFHELII stub.  
 Since a phase is comprised of modules, use the LNKEDT listing to identify the failing module.

-----  
**PHASE CICSTST2, \***  
 -----

**INCLUDE DFHELII**

```
** MODULE DFHELII 1999-02-04 11.27 INCLUDED FROM PRD1. BASE
** MODULE CEEBETEL 1999-02-04 11.32 AUTOLNKD FROM PRD2. SCEEBASE
** MODULE CEEBINT 1999-02-04 11.32 AUTOLNKD FROM PRD2. SCEEBASE
```

•  
•  
•

**ENTRY**

03/25/1999	PHASE	XFR-AD	LOCORE	HICORE	CSECT/ ENTRY	LOADED AT	TAKEN FROM	RMODE/ FROM
	CICSTST2	600098	600078	601BC9				31

Phase + x'5EE' is at  
 COBTEST + x'5CE'

<b>DFHELII</b>	<b>600078</b>	<b>DFHELII</b>	<b>31</b>
*DFHEPIN	600078		
*DFHEXEC	600080		
<b>COBTEST</b>	<b>600098</b>	<b>SYSLNK</b>	<b>ANY</b>
<b>CEEBETEL</b>	<b>600C88</b>	<b>CEEBETEL</b>	<b>ANY</b>
<b>CEEBINT</b>	<b>600CAS</b>	<b>CEEBINT</b>	<b>ANY</b>

## Correlate VSE/ESA Dump with COBOL Source

•  
•  
•

**UNRESOLVED EXTERNAL REFERENCES**

	<b>WKTRN</b>	<b>CEEUOPT</b>
	<b>WKTRN</b>	<b>CEERXITA</b>
	<b>WKTRN</b>	<b>CEESG000</b>
	<b>WKTRN</b>	<b>CEESG001</b>
	<b>WKTRN</b>	<b>CEESG002</b>

•  
•  
•

**UNRESOLVED ADCON AT OFFSET 00600C98**  
**UNRESOLVED ADCON AT OFFSET 00600C8C**  
**UNRESOLVED ADCON AT OFFSET 00600CC0**  
**UNRESOLVED ADCON AT OFFSET 00600CC4**

•  
•  
•

**024 UNRESOLVED ADDRESS CONSTANTS**  
**PHASE(S) CATALOGED INTO SUBLIBRARY PRIVB1 . OLSEN VOLID= PACC89**  
**1S55I LAST RETURN CODE WAS 0002**  
**EOJ TEST2CMP MAX. RETURN CODE=0002** **DATE 03/25/1999,**

RC=02 is normal for a  
COBOL/VSE link-edit.

# Correlate VSE/ESA Dump with COBOL Source

## 6. Locate this displacement in COBOL Listing

```
// JOB TEST2CMP COMPILE PROGRAM                                DATE 03/25/1999,
// LIBDEF *, SEARCH=(PRIVB1. OLSEN, PRD2. SCEECICS, PRD2. SCEERASE, PRD2. PROD)
// LIBDEF PHASE, CATALOG=PRIVB1. OLSEN
// OPTION ERRS, SXREF, SYM CATAL, NODECK
// PHASE CICSTST2, *
// INCLUDE DFHELII
// EXEC IGYCRCTL, SIZE=IGYCRCTL
PP 5686-068 IEM COBOL FOR VSE/ESA 1.1.0                        09/13/1994                DATE
JCL OPTION PARAMETERS:
NODECK, LINK, LIST, NOLISTX, SYM NOTERM, SXREF
PROCESS(CBL) STATEMENTS:
  CBL LIB, APOST, NOADV, NODYNAM RENT, BUF(4096), LIST
  CBL RENT, NODYNAM LIB
LINEID  PL  SL  ----- *A-1-B-+----- 2----- 3----- 4----- 5----- 6--
000001                                     IDENTIFICATION DIVISION.
000002                                     PROGRAM ID.                                COBTST.
000003                                     AUTHDR.                                CHARLES E. OLSEN
000007                                     ENVIRONMENT DIVISION.
000008                                     INPUT-OUTPUT SECTION.
000009                                     FILE-CONTROL.
000015                                     DATA DIVISION.
000021                                     WORKING-STORAGE SECTION.
.
.
.
```

Identify the COBOL statement which is at displacement x'5CE'

## Correlate VSE/ESA Dump with COBOL Source

```
000031      01  Starting-Message.
000032          05                      PIC X(31)
000033              VALUE 'This program contains a loop.'.
000034          05                      PIC X(30)
000035              VALUE 'Press enter to begin loop. '.
000036      77  MSGAREA                      PIC X(80).
000037      77  Console-Inarea              PIC X(200).
000038      77  VSAM Key                     PIC X(10) VALUE ...
000097  LINKAGE SECTION.
000098      01  DFHEIBLK
      .
      .
000130      01  DFHCOMMAREA PICTURE X(1).
000131  PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
000132      MOVE 2 TO Console-Record-Length.
      .
      .
000153  LOOP.
000154      MOVE VSAM Base-Record TO MSGAREA.
000155      MOVE 80 TO Console-Record-Length.
000156      GO TO LOOP.
      .
      .
000258      EQJ.
000259      GOBACK.
      .
      .
```

Program loop at  
statements 153-156

## Correlate VSE/ESA Dump with COBOL Source

PP 5686-068	IBM COBOL FOR VSE/ESA 1.1.0	09/13/1994	COBTEST	DATE
000000	COBTEST DS OH			
	USING *,15			
000000	47F0 F028	B 40(,15)		BYPASS CONSTANTS. BRANCH TO
000004	00	DC AL1(0)		ZERO NAME LENGTH FOR DUMPS
000005	C3C5C5	DC CL3'CEE'		CEE EYE CATCHER
000008	00000000	DC F'0'		STACK FRAME SIZE
00000C	00000014	DC A(@PPA1-COBTEST)		OFFSET TO PPA1 FROM ENTR
000010	47F0 F001			
.				
.				
.				
000060	C3D6C2E3C5E2E340	DC C' <b>COBTEST</b> '		PROGRAM NAME
000068	F1F9F9F9	DC CL4' <b>1999</b> '	@TIMEVRS:	YEAR OF COMPILATION
00006C	F0F3F2F5	DC CL4' <b>0325</b> '		MONTH/DAY OF COMPIL
000070	F1F6F1F9	DC CL4' <b>1619</b> '		HOURS/MINUTES OF CO
000074	F2F2	DC CL2' <b>22</b> '		SECONDS FOR COMPILA
000076	F0F1F0F1F0F0	DC CL6' 010100'		VERSION/RELEASE/MDD LEVEL 0
.				
.				
.				
<b>PROGRAM GLOBAL TABLE BEGINS AT LOCATION 0000F4 FOR 000030 BYTES</b>				
<b>LITERAL POOL MAP FOR SYSTEM LITERALS IN</b>				
0000F4 (LIT+0)	00000000	40404040	40404040	40404040 40404040 40404040 ...
000114 (LIT+32)	40404000	00000C		
000130 (LIT+0)	00000001	00000000	0050000D	000200C8 E28896A4 93844095 ...
			85838589	<sup>3</sup> .....&.....HShould not recei <sup>3</sup>
000150 (LIT+32)	A58540A3	8889A240	85999996	99404DC9 93939687 898340C5 ...
			5D407E7E	<sup>3</sup> ve this error (Illogic Error) == <sup>3</sup>

LE/VSE preface = x'F4' bytes  
Set up and call CEESTART

Literal Pool

## Correlate VSE/ESA Dump with COBOL Source

```

000170 (LIT+64) 6ED58194 85409596 A3408995 40C995A3 8599A2A8 A2A38594 ...
                                     9385407E ³>Name not in Intersystem Table =³
.
.
.
0003D4 5820 D130          L    2, 304(0, 13)          BL=1
0003D8 D203 2000 A010     MWC  0(4, 2), 16(10)        SORT-CORE-SIZE
0003DE D203 2008 A010     MWC  8(4, 2), 16(10)        SORT-FILE-SIZE
0003E4 D203 2010 A010     MWC 16(4, 2), 16(10)       SORT-MDE-SIZE
.
.
.
0005C0 58B0 C02C          L    11, 44(0, 12)         PRL=1
0005C4 47F0 B46A          BC   15, 1130(0, 11)       NO-RECORD-FOUND
0005C8                      GN=23  EQU  *
000153 *LOOP
000154 MOVE
0005C8 D24F 9110 9000     MWC  272(80, 9), 0(9)      MSGAREA
000155 MOVE
0005CE D201 9240 A014     MWC  576(2, 9), 20(10)    CONSOLE-RECORD-
000156 GO
0005D4 47F0 B1F4          BC   15, 500(0, 11)       LOOP

```

Start of Code (where base reg was pointing)

Loop: Statements 153 – 156  
Program Displacement +x'5CE'

To summarize:

1. Loop occurred at address x'01B005EE'.
2. By locating the eyecatcher, we determined this was +x'5EE' past DFHELII (into the phase).
3. From the link-edit listing, this is COBTEST + x'5CE'
4. COBOL listing shows this to be statement 155.



# Stand-alone Dump to Disk

## Symptom:

Stand-alone dumps tapes:

- Take too long
- Can be hard to find in an emergency
- Require on-site operations

## Resolution:

A Stand-alone dump may be taken to dasd:

- Quicker.
- More convenient.
- Remote access.

## Disadvantages:

- Previous dump must be processed before a new dump may be taken.

# Stand-alone Dump to Disk

## Speaker Notes:

A Stand-alone dump is taken when VSE/ESA has suffered a catastrophic error, and IBM support may require a snapshot of the entire VSE/ESA system. Sometimes it is also necessary to artificially freeze the system at a failure point for diagnosis.

Most Stand-alone dumps are taken to tape. This requires, however, that the operator have a tape available when the error occurs, and knows how to use it. In addition, system outages always occur at the worst time, usually when the system is fully loaded and under stress. Operations may be reluctant to take the time for a Stand-alone dump prior to bringing the system back up.

Systems without off-shift operations support provide unique challenges for system diagnosis. While it is possible to re-ipl a VSE/ESA system without on-site operational support, at most customer sites to use a tape, someone must mount it. Taking a stand-alone dump to disk allows the abend documentation to be analyzed and/or sent to Level2 support without requiring a physical tape.

## **Resolution:**

A Stand-alone dump may be taken to either tape or dasd. The advantages of using dasd as the medium for Stand-alone dump are:

1. **Quicker.** In a “system down” situation, a stand-alone dump to disk takes approximately 1/3 as much time as a stand-alone dump to tape.
2. **More convenient.** Operator does not have to locate the stand-alone dump tape in a crisis situation.
3. **Remote access.** Processing the dump after the system has been re-ipl'ed does not require additional tape mounts.

## **Disadvantages:**

1. Stand-alone dump files on disk are fully reusable, and, just like tape, taking a second stand-alone dump will overlay the previous dump files. Therefore, you must either have multiple dasd addresses prepared ahead of time, or process / offload the previous dump prior to taking a new stand-alone dump.

# Stand-alone Dump to Disk

## Steps to follow:

1. Define Stand-alone dump on dasd
2. Add labels to standard labels, and assigns to CICS start-up job
3. Take a trial stand-alone dump
4. Check, using DITTO, to see how much space you really used
5. Scan dump file using DOSVSDMP to ensure that all critical partitions are being dumped
6. Test procedure to onload dump into VSEDUMP (InfoAnal) library
7. Test procedure to offload dump to tape for transmission to VSE Level2.
8. Ensure Operations understands the procedures when and how to take a stand-alone dump.
9. Run with **DEBUG ON**.

# Stand-alone Dump to Disk

## Speaker Notes:

### **Stand-alone dumps on dasd can also be processed using the Interactive Interface**

(4.3:“Storage Dump Management”). However, this requires that the stand-alone dump extent on disk must be created using the Interactive Interface (4.6.2. “Create Stand-alone Dump Program on Disk”)

### **1. Define Stand-alone dump on dasd:** (See “VSE/ESA Diagnosis Tools”, Chapter 3)

CICS TS	(30 Meg)
POWER	(1.6 Meg)
VTAM	(7 Meg)
Supervisor and SVA	(12 Meg)
Total:	(50.6 Meg)

Three partitions on our test system were brought up with “// OPTION SADUMP=5”, indicating they should be included in the stand-alone dump file. Partition sizes indicate a stand-alone dump size of over 50meg. However, dump records are not written for unused pages.

### **Create Stand-alone dump files on disk:**

```
// JOB DOSVSDMP
// DLBL IJSYSDI, ' VSEDUMP. PROGRAM ON. DISK' , 99/365, SD
// EXTENT SYS013, SYSWK2, 1, 0, 1, 7
// DLBL IJSYSDU, ' STANDALONE. DUMP. ON. DISK' , 99/365, SD
// EXTENT SYS013, SYSWK2, 1, 0, 8, 7000
// ASSGN SYS013, DISK, VOL=SYSWK2, SHR
// EXEC DOSVSDMP, PARM=' CREATE DUMP DEVICE=SYS013'
/*
/ &
```

```
BG 0000 // JOB DOSVSDMP
          DATE 03/13/2000, CLOCK 22/57/46
BG 0000 4G09I DUMP PROGRAM HAS BEEN CREATED
BG 0000 4G27I DUMP FILE CAPACITY IS 72000 K BYTES
BG 0000 EOJ DOSVSDMP MAX. RETURN CODE=0008
          DATE 03/13/2000, CLOCK 23/02/31, DURATION 00/04/45
```

# Stand-alone Dump to Disk

## 2. Add labels to standard labels, and assigns to CICS start-up job

If you want to use Interactive Interface to create and manage the Stand-alone dump file

```
CATALOG DTRINFOA. PROC      EOD=YY DATA=YES REPLACE=YES
// ASSGN SYS016, DISK, VOL=SYSWK1, SHR      INFO ANAL MANAGEMENT FILE
// ASSGN SYS017, DISK, VOL=SYSWK1, SHR      INFO ANAL ROUTINES FILE
// ASSGN SYS013, DISK, VOL=SYSWK2, SHR      STAND-ALONE DUMP ←
YY
```

## 3. Take a trial stand-alone dump

```
* CP I 142
16: 10: 52 * MSG FROM VSETEST3: 4G34I STAND-ALONE DUMP IN PROGRESS ON DISK 142
16: 11: 29 * MSG FROM VSETEST3: 4G10I STAND-ALONE DUMP COMPLETE
HCPGIR450W CP entered; disabled wait PSW 000A0000 00CE0000
```

Stand-alone dump to disk took 35 seconds, versus 90 seconds to tape. This was a minimal test system. Obviously, a fully-functional production system will take much longer.

**\*\* Be sure to close your on-line files before you try this step. Doing a system reset with on-line files still open can cause corruption in VSAM and DL/I files. \*\***

## 4. Check, using DITTO, to see how much space you really used

DOSVSDMP pre-formats the dasd extent with 4112-byte records beginning with x'00000008' followed by three words of zero, and '\*\*\*\*\* UNUSED \*\*\*\*\*' at displacement x'10'. (See sample listings at the end of this section).

```
// JOB SCAN DISK
// ASSGN SYS013, DISK, VOL=SYSWK3, SHR
// UPSI 1
// EXEC DITTO
$$DITTO DRS INPUT=SYS013, BEGIN=000008, END=046714, SCANTYPE=D,
$$DITTO      SCANPOS=017, SCANARG='***** UNUSED *****'
$$DITTO EOJ
/*
/&
```

## Stand-alone Dump to Disk

### 5. Scan dump file using DOSVSDMP to ensure that all critical partitions are being dumped

All long-running jobs (CICS, POWER, VTAM, SQL, Vendor products) should be included in a stand-alone dump.  
Add “// OPTION SADUMP=nn” to the start-up jobs of any missing applications.

```
// EXEC DOSVSDMP, PARM='SCAN DEVICE=SYS013'
```

```
DIRECTORY OF VSE DUMP DATA SET
```

DUMP FILE	DUMP TYPE	NAME	DATE	DATA DUMPED
001	SADUMP			SUPERVISOR+SVA
002	SADUMP		00/03/14	PMRAS- R
003	SADUMP		00/03/14	PMRAS- 00
004	SADUMP	VTAMSTRT	00/03/14	F3- PARTITION
005	SADUMP	CICSI CCF	00/03/14	F2- PARTITION
006	SADUMP	POWSTART	00/03/14	F1- PARTITION

```
END OF DUMP
```

```
EOJ DOSVSDMP
```

```
DATE 03/15/2000, CLOCK 11/16/59, DURAT
```

### 6. Test procedure to onload dump into VSEDUMP (InfoAnal) library

Each dump must be loaded into the VSEDUMP library in a separate INFOANAL step.

```
// JOB INFOANAL ONLOAD S/A DUMP FROM DISK
```

```
// EXEC PROC=DTRINFOA
```

```
// EXEC INFOANA, SIZE=300K
```

```
SELECT DUMP MANAGEMENT
```

```
*** DELETE PREVIOUS DUMPS ***
```

```
DUMP NAME SYSDUMP. BG. SAD00001
```

```
DELETE
```

```
DUMP NAME SYSDUMP. F2. SAD00001
```

```
DELETE
```

```
RETURN
```

```
DUMP NAME SYSDUMP. BG. SAD00001
```

```
*** Load Supervisor and SVA ***
```

```
SELECT DUMP ONLOAD
```

```
VOLID DISK SYS013
```

```
FILE 1
```

```
←
```

```
RETURN
```

## Stand-alone Dump to Disk

```
DUMP NAME SYSDUMP. F2. SAD00001 *** Load CICS partition ***  
SELECT DUMP ONLOAD  
VOLID DISK SYS013  
FILE 5 LAST ←  
RETURN  
SELECT END  
/*  
/&
```

### 7. Test procedure to offload dump to tape for transmission to VSE Level2.

```
// JOB DITTO - COPY STAND-ALONE DUMP TO TAPE  
// UPSI 1  
// ASSGN SYS013, DISK, VOL=SYSWK2, SHR  
// EXEC DITTO, SIZE=512K  
$$DITTO REW OUTPUT=181  
$$DITTO ST INPUT=SYS013, FILEIN=IJSYSDU, OUTPUT=181, RECFMOUT=F  
/*  
/&
```

### 8. Ensure Operations understands the procedures when and how to take a stand-alone dump.

Actual procedures will depend on the processor type and hardware console configuration.

1. Display low core and write down or screen print the first x'100' bytes. Stand-alone dump will destroy several low core locations which may be critical for later problem determination.
2. Perform "STORE STATUS". This stores the current PSW and registers into a pre-defined location in low core.
3. It is not necessary to write down recent error messages on the system console. These can be retrieved from the console buffers in the stand-alone dump.
4. Set the IPL device to the stand-alone dump medium. Mount tape, if necessary
5. Set "CLEAR" to "NO", and IPL the stand-alone tape / dasd

## Stand-alone Dump to Disk

This page  
left  
intentionally  
blank



# Stand-alone Dump to Disk

## Additional InfoAnal jobs:

It is necessary to load the data from the stand-alone dump files on disk into a VSEDUMP library, prior to formatting the individual partition dumps, or analyzing them. The dump may be offloaded directly to tape without copying it into a VSEDUMP library. Let Level2 know that this is a DITTO tape backup of a stand-alone dump taken to dasd.

## Format and print diagnostic information:

```
// JOB INFOANAL ANALYZE S/A DUMP
// EXEC PROC=DTRINFOA
// EXEC INFOANA, SIZE=300K
  SELECT DUMP MANAGEMENT
    DUMP NAME SYSDUMP.BG.SAD00001   *** Run against Supv / SVA dump ***
    RETURN
  SELECT DUMP VIEWING
    CALL IJBXDEBUG   *** Analyze Stand-alone dump ***
                    *** Ignore msgBLN9003I 4G80I ***
    CALL IJBXSDA     *** Format In-core SDAID Buffer ***
    CALL IJBXCSMG   *** Print Console Buffer ***
    RETURN
  SELECT DUMP VIEWING
    PRINT FORMAT    *** Print LBD entries ***
  SELECT END
/*
/ &
```

IJBXDEBUG creates an LBD entry called DBUGHDR. This is printed out by the PRINT FORMAT, along with other dump information and symptom records created during dump creation.

# Stand-alone Dump to Disk

Offload dump to tape for transmission to IBM Support:

```
// JOB OFFLOAD DUMPS FROM VSEDUMP LIBRARY
// ASSGN SYS009, 181
// MTC REW, 181
// EXEC PROC=DTRINFOA
// EXEC INFOANA, SIZE=300K
DUMP NAME SYSDUMP. BG. SAD00001 *** Copy Supv / SVA dump to tape ***
SELECT DUMP OFFLOAD
VOLID SADUMP SYS009
ERASE NO
RETURN
DUMP NAME SYSDUMP. F2. SAD00001 *** Stack additional dumps to tape ***
SELECT DUMP OFFLOAD
VOLID SADUMP SYS009
ERASE NO
RETURN
SELECT END
/*
/ &
```

```
// JOB DITTO - PRINT DISK EXTENTS
*
* ADDRESS IS CCCHH IN DECIMAL. CYLINDER AND HEAD BOTH START WITH ZERO.
*
// UPSI 1
// ASSGN SYS010, DISK, VOL=SYSWK2, SHR
// EXEC DITTO, SIZE=512K
$$DITTO SET PAGESKIP=NO, DATAHDR=NO, DUMP=ACROSS
$$DITTO DP FORMAT=HEX, INPUT=SYS010, BEGIN=03209, END=03514
/*
/ &
```

# Format of Stand-alone Dump on Disk

DOSVSDMP pre-formats the dasd extent with 4112-byte records beginning with x'00000008' followed by three words of zero:

```

DITTO/ESA for VSE
$$DITTO SET PAGESKIP=NO, DATAHDR=NO, DUMP=ACROSS
$$DITTO DP FORMAT=HEX, INPUT=SYS010, BEGIN=00100, END=00101
* * * * Device 0142, 3390, VOLSER=SYSWK2, Cylinder 0, Head 8, operative primary track
CYL-HD- REC 00000- 08- 001 DATA LEN 4112
000000 00000008 00000000 00000000 00000000 5C5C5C5C 40E4D5E4 E2C5C440 5C5C5C5C *          **** UNUSED ****
000020 5B5BC2D6 D7C5D540 5B5BC2C3 D3D6E2C5 5BC9D1C2 C1E2C7D5 F1404040 F0F0F0F0 *$$BOPEN $$BCLOSESIJBASGN1 0000*
000040 C5D5C440 00000008 00400650 F0F0F0F0 0040019C 0000001C 00000010 00000004 *END          &0000          *
000060 00000008 00000018 00400E9A 00400E9C 00400E9D 0001D940 F140F240 F340F440 *          R 1 2 3 4 *
    
```

After S/A dump, dasd file looks like this:

- Records beginning with x'00000001' are Symptom / LBD records
- Records beginning with x'00000002' are a 4K storage block. Word 2 is the virtual address; word 3 is the real address. Hex data begins +x'10'.
- Dump for next partitions is preceded by one or more x'00000001' record(s).

```

$$DITTO SET PAGESKIP=NO, DATAHDR=NO, DUMP=ACROSS
$$DITTO DP FORMAT=HEX, INPUT=SYS010, BEGIN=00008, END=00008
* * * * Device 0142, 3390, VOLSER=SYSWK2, Cylinder 0, Head 8, operative primary track
CYL-HD- REC 00000- 08- 001 DATA LEN 4112
000000 00000001 E2D9F9F6 F7F2F0F4 F9F4F1F0 B3BDFC75 9C1A6644 00000000 00000000 * SR9672049410 *
000020 00000000 00000000 00000000 0000F5F6 F8F6F0F6 F6F0F6F4 F5C34080 E2C1C4E4 *          56860660645C SADU*
000040 D4D74040 00000000 00000000 00000074 00000074 00A60074 0010011A 00000000 *MP          w          *
000060 00000000 00000000 00000000 00000000 00000000 00000000 D4C1C3C8 C9D5C56D *          MACHINE_ *
000080 E2E3C1E3 E4E26DD5 D6E36DE2 C1E5C5C4 40C4C1E3 C56DD5D6 E36DC1E5 C1C9D3C1 *STATUS_NOT_SAVED DATE_NOT_AVAILA*
0000A0 C2D3C540 D4C1C3C8 C9D5C57E C5E2C140 D4D6C4C5 7ED7C1C7 C9D5C740 C1C3E3C9 *BLE MACHINE=ESA MODE=PAGING ACTI*
0000C0 E5C56DE2 D7C1C3C5 6DC9C47E E24040C4 E4D4D7C5 C46DC4C1 E3C16DC6 D9D6D46D *VE_SPACE_ID=S DUMPED_DATA_FROM_ *
0000E0 E2D7C1C3 C56DC9C4 7EE24040 D7D4D96D C1C4C4D9 C5E2E26D E2D7C1C3 C56DC9C4 *SPACE_ID=S PMR_ADDRESS_SPACE_ID*
000100 7EF0F040 C4E4D4D7 C5C46DC4 C1E3C17E E2E4D7C5 D9E5C9E2 D6D94EE2 E5C10000 *=00 DUMPED_DATA=SUPERVISOR+SVA *
000120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * *
000140 to 00100F same as previous line
CYL-HD- REC 00000- 08- 002 DATA LEN 4112
000000 00000002 00000000 00000000 061AF1F0 00080000 00000910 06000300 70FF0090 *          10          *
CYL-HD- REC 00000- 08- 003 DATA LEN 4112
000000 00000002 00001000 00001000 061A0000 00000000 00000000 00000000 00000000 * *
CYL-HD- REC 00000- 08- 004 DATA LEN 4112
000000 00000002 00002000 00002000 061A0000 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF * *
CYL-HD- REC 00000- 08- 005 DATA LEN 4112
000000 00000002 00003000 00003000 061A0000 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF * *
    
```

## Format of Stand-alone Dump on Disk

```

CYL-HD- REC 00032- 12- 010 DATA LEN 4112
000000 00000002 021FF000 017DE000 061A0000 021FEFEC 000046A0 FFFF0100 021FF01C * 0 ' 0 *
CYL-HD- REC 00032- 12- 011 DATA LEN 4112
000000 00000001 E2C3E3F6 00000000 00000000 00000000 D3C2C440 002C0000 C6F24040 * SCT6 LBD F2 *
CYL-HD- REC 00032- 12- 012 DATA LEN 4112
000000 00000004 00000000 00000000 00000000 021FEFEC 000046A0 FFFF0100 021FF01C * 0 *
CYL-HD- REC 00032- 13- 001 DATA LEN 4112
000000 00000001 E2D9F9F6 F7F2F0F4 F9F4F1F0 B3BDFC75 9C1A6644 000000F0 F061F0F3 * SR9672049410 00/03*
000020 61F1F400 00000000 00000000 0000F5F6 F8F6F0F6 F6F0F6F4 F5C34080 E2C1C4E4 */14 56860660645C SADU*
000040 D4D74040 00000000 00000000 00000074 00000074 00B10074 00100125 00000000 *MP *
000060 00000000 00000000 00000000 00000000 00000000 00000000 D4C1C3C8 C9D5C56D * MACHINE_*
000080 E2E3C1E3 E4E26DD5 D6E36DE2 C1E5C5C4 40C4C1E3 C57EF0F0 61F0F361 F1F440D4 *STATUS_NOT_SAVED DATE=00/03/14 M*
0000A0 C1C3C8C9 D5C57EC5 E2C140D4 D6C4C57E D7C1C7C9 D5C740C1 C3E3C9E5 C56DE2D7 *ACHINE=ESA MDDE=PAGING ACTIVE_SP*
0000C0 C1C3C56D C9C47EE2 4040C4E4 D4D7C5C4 6DC4C1E3 C16DC6D9 D6D46DE2 D7C1C3C5 *ACE_ID=S DUMPED_DATA_FROM_SPACE*
0000E0 6DC9C47E F14040D7 D4D96DC1 C4C4D9C5 E2E26DE2 D7C1C3C5 6DC9C47E F0F040C4 *_ID=1 PMR_ADDRESS_SPACE_ID=00 D*
000100 E4D4D7C5 C46DC4C1 E3C17EC6 F160D7C1 D9E3C9E3 C9D6D540 D1D6C26D D5C1D4C5 *UMPED_DATA=F1- PARTITION JOB_NAME*
000120 7ED7D6E6 E2E3C1D9 E3000000 00000000 00000000 00000000 00000000 00000000 *=POWSTART *
000140 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * *
000160 to 00100F same as previous line
CYL-HD- REC 00032- 13- 002 DATA LEN 4112
000000 00000002 00400000 00B9E000 C61A0000 C9D7E6D7 D6E6C5D9 070D0000 0029EB16 * F IPWPOWER *
CYL-HD- REC 00032- 13- 003 DATA LEN 4112
000000 00000002 00401000 00BB0000 C61A0000 E2E3C1D9 E340D6C6 40C6C9E7 C1C2D3C5 * F START OF FIXABLE*
.
.
CYL-HD- REC 00034- 03- 001 DATA LEN 4112
000000 00000002 0058F000 00BA0000 061A0000 00000000 FFFFFFFF FFFFFFFF 00000000 * 0 *
CYL-HD- REC 00034- 03- 002 DATA LEN 4112
000000 00000001 E2C3E3F6 00000000 00000000 00000000 D3C2C440 002C0000 C6F14040 * SCT6 LBD F1 *
CYL-HD- REC 00034- 03- 003 DATA LEN 4112
000000 00000004 00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF 00000000 * *
CYL-HD- REC 00034- 03- 004 DATA LEN 4112
000000 00000008 00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF 00000000 * *
CYL-HD- REC 00034- 03- 005 DATA LEN 4112
000000 00000008 00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF 00000000 * *
CYL-HD- REC 00034- 03- 006 DATA LEN 4112
000000 00000008 00000000 00000000 00000000 5C5C5C5C 40E4D5E4 E2C5C440 5C5C5C5C * **** UNUSED ****

```

This record format is the same as for a stand-alone dump on tape, and an AR DUMP to tape,  
as well as for the internal dump format within the VSEDUMP library.  
The only difference is that for a stand-alone dump on tape, each partition is in a separate tape file.

# Sharing Physical Tape Units

## Symptom:

0P03I Device=CUU in use by other system

## Resolution:

1. **DVCDN / DVDUP** or **OFFLINE / ONLINE** are not necessary.
2. JCL **ASSGN** statement generates **ASSIGN CCW** command
3. Assignment is released (at end-of-job, or via the **ASSGN <cuu>,UA** command)
4. **OFFLINE <cuu>**
5. **ONLINE <cuu>,HOLD ==>** no **UNASSIGN CCW** command.
6. **VOLUME** or **VOLUME TAPE**
7. **NOASSGN:**
8. **NOAVR**

# Sharing Physical Tape Units

## Speaker Notes:

Customer is sharing tape drives between two or more VSE/ESA systems. Occasionally he (she?) receives error message “*OP03I Device=<cuu> in use by other system*”.

Generally this problem is caused by a permanent assignment of a logical unit to the tape on the owning system. This prevents VSE/ESA from automatically releasing ownership of the tape when processing is finished.

VSE/ESA supports a tape drive assigned to multiple physical VSE/ESA systems. It is not possible to assign a physical device to multiple virtual machines under VM/ESA. VSE/ESA controls device access using the ASSIGN / UNASSIGN CCW commands (x'B7' / x'C7'), which are supported on most “modern” tape drives (3480 / 3490). These commands are similar to the dasd RESERVE / RELEASE commands.

1. It is generally not necessary to use the **DVCDN / DVDUP** or **OFFLINE / ONLINE** commands to control tape unit assignment.
2. Whenever VSE/ESA encounters a JCL **ASSGN** statement for a tape unit, it issues an ASSIGN CCW command to that drive.
3. Generally, when the assignment is released (at end-of-job, or via the **ASSGN <cuu>,UA** command) VSE/ESA issues an UNASSIGN CCW command to the device.
4. **OFFLINE <cuu>**: If a tape unit remains assigned to a VSE/ESA system even after all assignments have apparently been released, the **OFFLINE <cuu>** command will unconditionally release all assignments (including permanent assignments) and then issue an UNASSIGN CCW command to the tape device.
5. The **ONLINE <cuu>,HOLD** command will instruct VSE/ESA not to issue the UNASSIGN CCW command, and the tape unit will remain assigned to this VSE/ESA system until released by the **OFFLINE** command.
6. The **VOLUME** or **VOLUME TAPE** command will show the status (including valid) of all tape units accessible to this VSE/ESA system, but will not issue an ASSIGN CCW command. Thus, the **VOLUME** command may be used on multiple VSE/ESA systems. If it is issued at precisely the same time, one system may receive a “device busy” error, but re-issuing the command should work ok.
7. **NOASSGN**: This undocumented AR command was provided in the early 3480 days because of vendor product dependencies. This command prevents all ASSIGN / UNASSIGN CCW commands from being issued to tape units. Thus, any VSE/ESA system can write to any tape unit at any time, with the resultant danger of file contention / corruption. This command should only be used in the most dire of circumstances. “**NOASSGN OFF**” turns the “**ASSIGN / UNASSIGN**” process back on.
8. **NOAVR** command: When a cartridge is inserted into a drive and the tape is made ready, an interrupt is generated to all VSE/ESA systems physically connected to that drive. The VSE/ESA system will then initiate “Automatic Volume Recognition” processing and read the tape label. This will later be displayed by the **STATUS** command. Sometimes this “AVR” processing interferes with vendor products who are also doing device sensing and volume label management. The AR **NOAVR** command may be used to tell VSE/ESA to skip automatic volume sensing. The output of the **STATUS** command will be unpredictable. Tape **ASSIGN** or **UNASSIGN CCW** commands are not issued during AVR processing. “**NOAVR OFF**” will reinitiate “AVR” processing.

# Installing Service from disk

## Symptom:

Customer has received PTFs electronically, but cannot apply them serially due to internal co-requisites.

## Resolution:

1. Copy PTFs to dasd using following jcl:

```
// JOB DITTO - COPY CARD (PTFS) TO DISK
// UPSI 1
// ASSGN SYS013, DISK, VOL=SYSWK2, SHR
// DLRL IJSYSPF, ' PTF. FILES. ON. DISK' , 0, SD
// EXTENT SYS013, SYSWK2, 1, 0, 15, 225
// EXEC DITTO, SIZE=512K
$$DITTO SET EOD=$$$$$$$$
$$DITTO CS OUTPUT=SYS013, FILEOUT=IJSYSPF, RECFMOUT=FB, BLKSIZE=3440
// JOB UD51226
// OPTION CATAL
* COMPONENT: 5686-066-05(06645C)
* APARS FIXED: DY45336
* SPECIAL CONDITIONS:
*   COPYRIGHT:           (C) COPYRIGHT IBM CORP. 1998
*   LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
* COMMENTS:
*   •
*   • Reminder of PTF stack
*   •
$$$$$$$$
/*
/ &
```

## Installing Service from disk

2. Apply PTFs from disk using following jcl:

```
// JOB MSHP  
// ASSGN SYS013, DISK, VOL=SYSWK2, SHR  
// DLBL IJSYSPF, ' PTF. FILES. ON. DISK' , 0, SD  
// EXTENT SYS013, SYSWK2, 1, 0, 15, 225  
// EXEC MSHP  
INSTALL SERVICE FROMDISK  
/*  
/&
```



## PD/PSI Console Commands

### ALTER { |space\_id | syslogid} ,address

See “*VSE/ESA System Control Statements*” (SC33-6613)

Allows alteration of 16 bytes of storage currently resident (paged-in). Storage can be in supervisor, SVA, or problem program area (including CICS / VTAM / POWER).

```
DSPLY 1 , 300
AR 015 90F21028 18215811 00185801 0014F9F9          *.2.....99*
AR 015 1140I  READY
ALTER 1 , 300
AR 015 1142D  ADDRESS WITHIN SUPERVISOR OR SVA
AR+015
15 IGNORE
AR 015 OLD DATA:  90F21028 18215811 00185801 0014F9F9 *.2.....99*
AR 015 ENTER HEX DATA (1-16 BYTES)
AR+015
15 FFF2
AR 015 1140I
DSPLY 1 , 300
AR 015 FFF21028 18215811 00185801 0014F9F9          *.2.....99*
AR 015 1140I  READY
```

### DSPLY (Use SHOW)

## PD/PSI Console Commands

### CANCEL cuu (,FORCE)

If device I/O has not completed, and is preventing a partition from going to EOJ, the **FORCE** option will artificially flag the I/O as completed, and reset all operating system indicators.

**This option must be used only with extreme caution!**

### // EXEC DTRIATTN , PARM = 'cmd'

Executes an attention routine command from JCL.

Ideal for executing an **AR** command during IPL processing. Just add the appropriate statement to the \$0JCL.PROC (BG ASI procedure)

```
BG- 0000 // PAUSE
```

```
0 // EXEC DTRIATTN,PARM='MAP'
```

```
MAP
```

```
AR 0015  SPACE AREA      V- SIZE  GETVIS  V- ADDR  UNUSED NAME  
AR 0015   S   SUP      716K           0      $$SSUPX
```

```
•  
•  
•
```

# PD/PSI Console Commands

## DEBUG

See “*VSE/ESA Supervisor Diagnosis Reference*” (LY33-9164)

- Designed to trace supervisor events. Subject to change without notice.
- Offers Event Tracing and Address Compare Stop
- Can be tailored to only trace specific events.

DEBUG Displays current DEBUG trace setting.

DEBUG ON (,nnK)

DEBUG BG,ON (,nnK)

DEBUG OFF

DEBUG SUSPEND

DEBUG END

DEBUG TRACE=

DEBUG TRACE=NOSWCH

DEBUG SHOW

DEBUG PSHOW

DEBUG NSHOW

DEBUG STOP [,id],[+]addr.len,{EQ | NE | LO | HI},string

# PD/PSI Console Commands

## Speaker Notes:

**DEBUG** (with no options) Displays current DEBUG trace setting.

```
DEBUG
AR 0015 DEBUG ON
AR 0015 TRACE=PCK, TEST, REGS, TASK, INT, SIO, EXT, DISP, SVC, TERM, DATA, USER, TDTR
```

**DEBUG ON (,nnK)** Allocates three areas, and commences tracing. Default is 16K each, in 31-bit GETVIS.

**DEBUG BG,ON (,nnK)** Starts tracing only for BG partition

**DEBUG OFF** Temporarily suspends tracing. When reactivated (DEBUG ON) a trace area switch is performed.

**DEBUG SUSPEND** Temporarily suspends tracing. Does not switch trace area when reactivated (DEBUG ON).

**DEBUG END** Stops tracing, and releases trace areas.

**DEBUG TRACE=** {option,option} Specify events to be traced.

**DEBUG TRACE=NOSWCH** Turn off automatic debug buffer switch

**DEBUG SHOW** {=option,option} {,CUU=cuu} Format and print current trace area on console (or specified printer)

**DEBUG PSHOW** Format and print previous trace area.

**DEBUG NSHOW** Format and print next trace area.

**DEBUG STOP [,id],[+]addr.len,{EQ | NE | LO | HI},string**

Compare data at specified address for specified value, and set system into disabled wait (low core = x'0000EEEE'). Activating PSW RESTART, (VM, enter SYSTEM RESTART) will enable system to continue.

**DEBUG STOP,026C.4,EQ,0004EC40** Stop system when current PCB pointer (at low core x'26C, for a length of 4) points at the BG PCB (address x'04EC0').

# PD/PSI Console Commands

## DIAG

See “*VM/ESA CP Programming Services*” (SC24-5520)

Issues a VM/ESA Diagnose command.

Currently, only DIAG x'210' is supported, which returns virtual and real device and control unit information, along with data from a “Read Device Characteristics”.

## DOIO

**DOIO** — **cuu** — **, READ** — **, CYL=cchhr**  
**, WRITE** — **, CNT=cchhr**  
**, BLK=bbbbbbb**

**CYL** => Read record and type to console

**CNT** => Read record and type Count-Key-Data to console

**DOIO cuu, VOL1** type volume one label for specified device on console

**DOIO cuu, CMD=cc ff llll** => Perform command and type response on console

**cc** = CCW opcode

**ff** = Chain byte

**llll** = length

# PD/PSI Console Commands

## Speaker Notes:

```
DIAG 210,141
AR 0015
00000000 0141003A 04820000 04820A82 00000000 * . . . . . b . . . b . b . . . . *
00000010 3990EC33 900AD100 10962024 0458000F * . . . . . J . . o . . . . . *
00000020 E000E5A2 05940222 13090674 00000000 * \ . V s . m . . . . . *
00000030 00000000 00000000 2424 * . . . . . *
AR 0015 1I40I  READY
```

```
DOIO 120,CMD=04200020 ←== SENSE COMMAND
AR 0015  DATA WHICH WAS READ / PREPARED
00000000 00000000 13000000 000000AB 49000005 * . . . . . *
00000010 22006E31 70700000 00004CE2 00000000 * . . > . . . . . < S . . . . *
AR 0015 1I40I  READY
```

```
DOIO 120,READ,CYL=0023000D01 ←== Read a record from dasd
AR 0015  DATA WHICH WAS READ / PREPARED
00000000 1B224320 60004220 4018D258 6002A5AD * . . . . - . . . . . K . - . v . *
00000010 92FF6000 9120B114 4710FBD8 43210000 * k . - . j . . . . . Q . . . . *
00000020 95561000 4740FBC0 41200056 4420FC16 * n . . . . { . . . . . *
00000030 BDOCE02C 4740FBE8 BE0C6059 41200059 * . . \ . . . Y . . - . . . . *
AR 0015 1I40I  READY
```

```
DOIO 120,VOL1 ←== Read VOL1 label
AR 0015  DATA WHICH WAS READ / PREPARED
00000000 E5D6D3F1 C3E3E2F1 F2F04000 4A000E01 * VOL1CTS120 . c . . . *
00000010 00000005 0000004B 0000004B 00002BF2 * . . . . . 2 *
00000020 00000006 00000040 00000040 94032211 * . . . . . m . . . *
00000030 12540000 00000000 00000001 00000016 * . . . . . *
00000040 00000016 00000000 00000000 00000000 * . . . . . *
AR 0015 1I40I  READY
```

**PD/PSI Console Commands**

**FLT (Fetch Load Trace)**

```
FLT — syslogid [ ] , ON [ ]
                             [ ] , OFF [ ]
```

Creates an in-core Fetch / Load trace table.

**GETVIS**

See *“VSE/ESA 2.6 System Control Statements”* (SC33-6613)

```
GETVIS —————
           [ ]
           [ syslogid ]
           [ SVA ]
           [ ]
           [ ]
           [ ALL ]
           [ DETAIL ]
           [ RESET ]
```

Allows dynamic monitoring of Getvis usage.  
“ALL” shows GETVIS sub-pools at both the partition and SVA level.  
“DETAIL” also shows addresses of storage blocks allocated to each sub-pools.  
“RESET” will reset “Max Ever Used” values (partition only).

## PD/PSI Console Commands

### Speaker Notes:

1. If you are experiencing GETVIS shortages, which are not directly attributable to a specific application, issue “GETVIS ...,ALL” commands periodically during the day. GETVIS is normally acquired in an application-specific “sub-pool”, and if you identify the sub-pool which is growing, the name of the sub-pool can be used to identify the aberrant application.
2. During CICS TS startup, a GETVIS macro is issued for all of the storage available in the 24-bit area, then this storage is released. So, the MAX EVER USED value for the 24-bit area always showed the total storage available, not the high water mark ever used. “GETVIS F2,RESET” will reset the “MAX EVER USED” value back to the amount actually used, to allow more effective monitoring of 24-bit GETVIS acquisition.

<b>GETVIS F2</b>				
AR 0015	GETVIS USAGE	F2- 24	F2- ANY	
AR 0015	AREA SIZE:	<b>11, 260K</b>	30, 716K	
AR 0015	USED AREA:	8, 656K	25, 840K	MAX. EVER USED:
AR 0015	FREE AREA:	2, 604K	4, 876K	LARGEST FREE:
AR 0015	1140I	READY		
<b>GETVIS F2, RESET</b>				
AR 0015	GETVIS USAGE	F2- 24	F2- ANY	
AR 0015	AREA SIZE:	11, 260K	30, 716K	
AR 0015	USED AREA:	8, 656K	25, 840K	MAX. EVER USED:
AR 0015	FREE AREA:	2, 604K	4, 876K	LARGEST FREE:
AR 0015	1140I	READY		



## PD/PSI Console Commands

GETVIS					
AR 0015	GETVIS USAGE	SVA- 24	SVA- ANY	SVA- 24	SVA- ANY
AR 0015	AREA SIZE:	1, 372K	2, 580K		
AR 0015	USED AREA:	512K	1, 220K MAX. EVER USED:	528K	1, 240K
AR 0015	FREE AREA:	860K	1, 360K LARGEST FREE:	860K	860K
AR 0015	1I40I	READY			
GETVIS SVA					
AR 0015	GETVIS USAGE	SVA- 24	SVA- ANY	SVA- 24	SVA- ANY
AR 0015	AREA SIZE:	1, 372K	2, 580K		
AR 0015	USED AREA:	512K	1, 220K MAX. EVER USED:	528K	1, 240K
AR 0015	FREE AREA:	860K	1, 360K LARGEST FREE:	860K	860K
AR 0015	1I40I	READY			
GETVIS F4					
AR 0015	GETVIS USAGE	F2- 24	F2- ANY	F2- 24	F2- ANY
AR 0015	AREA SIZE:	4, 096K	4, 096K		
AR 0015	USED AREA:	3, 304K	3, 304K MAX. EVER USED:	3, 356K	3, 356K
AR 0015	FREE AREA:	792K	792K LARGEST FREE:	792K	792K
AR 0015	1I40I	READY			
GETVIS SVA,ALL					
AR 0015	GETVIS USAGE	SVA- 24	SVA- ANY	SVA- 24	SVA- ANY
AR 0015	AREA SIZE:	1, 432K	8, 740K		
AR 0015	USED AREA:	676K	1, 772K MAX. EVER USED:	684K	1, 780K
AR 0015	FREE AREA:	756K	6, 968K LARGEST FREE:	748K	6, 212K
AR 0015	SUMMARY REPORT				
AR 0015	SUBPOOL	REQUEST	<-- SVA- 24- AREA- - -	-- SVA- ANY- AREA- - >	
AR 0015	Default		204K	64K	
AR 0015	ISTSVF		80K	272K	
	:				
	:				
AR 0015	IEXT0011		OK	4K	
AR 0015	IJBCEM		OK	136K	
AR 0015	SUBPOOL TOTALS		660K	1, 004K	

## PD/PSI Console Commands

```

GETVIS SVA,DETAIL
AR 0015 GETVIS USAGE      SVA- 24      SVA- ANY      SVA- 24      SVA- ANY
AR 0015  AREA SIZE:      1, 432K      8, 740K
AR 0015  USED AREA:      676K      1, 772K MAX. EVER USED:  692K      1, 808K
AR 0015  FREE AREA:      756K      6, 968K LARGEST FREE:    748K      6, 212K
AR 0015 SUMMARY REPORT
AR 0015 SUBPOOL          REQUEST <-- SVA-24- AREA- - -      -- SVA- ANY- AREA- - - >
AR 0015 Default         204K      64K
AR 0015                  0023F000- 0023FFFF      02CBF000- 02CBFFFF
AR 0015                  00244000- 0024EFFF      02CB2000- 02CBDFFF
AR 0015                  00250000- 00250FFF      02C13000- 02C15FFF
AR 0015                  00252000- 00253FFF
AR 0015                  00255000- 00256FFF
AR 0015                  00258000- 00272FFF
AR 0015                  00275000- 00275FFF
AR 0015                  00278000- 00278FFF
AR 0015                  0027B000- 0027EFFF
AR 0015                  00281000- 00281FFF
AR 0015
AR 0015 I STSVF         80K      272K
AR 0015                  0029B000- 0029BFFF      02CA6000- 02CA7FFF
AR 0015                  0029B000- 0029BFFF      02CA6000- 02CA7FFF
AR 0015                  002A2000- 002A2FFF      02C91000- 02C91FFF
AR 0015                  002AB000- 002ABFFF      02C85000- 02C86FFF
AR 0015                  002B0000- 002B2FFF      02C7C000- 02C7EFFF
AR 0015                  002B6000- 002B6FFF      02C78000- 02C78FFF
AR 0015                  002B8000- 002B8FFF      02C69000- 02C6DFFF
AR 0015                  002D0000- 002D4FFF      02C1E000- 02C53FFF
AR 0015                  002D6000- 002DCFFF
AR 0015
AR 0015
AR 0015 I EXT0011      OK      4K
AR 0015                  02CDD000- 02CDDFFF
AR 0015 I JBCSM        OK      136K
AR 0015                  02CDE000- 02CFFFFF
AR 0015 SUBPOOL TOTALS 660K      1, 004K
AR 0015 1I40I  READY

```

## PD/PSI Console Commands

### IESINSRT

Punches JCL to allow multi-step jobs without using dasd as intermediate SYSPCH / SYSIPT.storage.

1. Punch all JCL until next \* \$\$ END
2. Convert #& to /&
3. Convert \$ \$\$ to \* \$\$ (POWER JECL)
4. Eliminates temporary dasd files and assignment changes to SYSPCH and SYSRDR/SYSIPT.

# PD/PSI Console Commands

```
* $$ JOB JNM=XXXXX, DISP=D, CLASS=A, NTFY=YES
* $$ LST DISP=D, CLASS=Q, PRI=3
* $$ PUN DISP=D BEST=*, PRI=9, CLASS=A
// JOB XXXXX TRANSLATE PROGRAM OTS1PGM
// ASSGN SYSIPT, SYSRDR
// EXEC IESINSRT
$ $$ LST DISP=D, CLASS=Q, PRI=3
// JOB XXXXX COMPILE PROGRAM OTS1PGM
// LIBDEF *, SEARCH=(PRD2. SCEECICS, PRD2. SCEEBASE, PRD2. TEST)
// LIBDEF PHASE, CATALOG=PRD2. TEST
// OPTION ERRS, SXREF, SYM, CATAL, NODECK
  PHASE OTS1PGM, *
INCLUDE DFHELI I
// EXEC IGYCRCTL, SIZE=IGYCRCTL
  CBL LIB, APOST, NOADV, NODYNAM, RENT, BUF(4096)
* $$ END
// ON $CANCEL OR $ABEND GOTO ENDJ2
// OPTION NOLIST, NODUMP, DECK
// EXEC DFHECP1$, SIZE=512K
  CBL XOPTS(COBOL2 CICS DEBUG)
000100 ID DIVISION.
000200*-----
000300*  DISPLAY MAPS1 ON TERMINAL
      .
      .
      .
001800  EXEC CICS RETURN TRANSID ('OTP1') END- EXEC.
001900  STOP RUN.
/*
/. ENDJ2
// EXEC IESINSRT
/*
// IF CATALOG NE 1 OR $MRC GT 4 THEN
// GOTO NOLNK
// EXEC LNKEDT, SIZE=256K
/. NOLNK
#&
$ $$ EOJ
* $$ END
/&
* $$ EOJ
```

Punch

Punch

Punches output  
back into POWER  
reader queue.

# PD/PSI Console Commands

## LOCATE

See “*VSE/ESA Supervisor Diagnosis Reference*” (LY33-9164)



' is the single, special character that must be used to indicate that the following string is a character string. If the ' is missing, the string is assumed to be hexadecimal digits.

string is the series which the user wants to be located; limited to either 16 characters or, to 32 hexadecimal digits representing 16 bytes of storage. Any character or hexadecimal digit to be excluded from the scan, must be replaced by a “.” (period).

## PD/PSI Console Commands

### Speaker Notes:

```

MAP
AR 0015 SPACE AREA          V- SIZE  GETVIS  V- ADDR  UNUSED NAME
AR 0015   S   SUP             640K           0           $$ASSUPX
AR 0015   S   SVA- 24        2304K    1600K    A0000       1600K
AR 0015   0   BG V           1280K    256K     600000      10752K

LOCATE 'IKQBFA (FROM=A0000,TO=600000
AR 0015 MATCH FOUND AT 001B2A2C
V001B2A20 ..... C9D2D8C2 06 *.....IKQB*
V001B2A30 C6C14040 F1F5C340 C4E8F4F3 F5F3F140 06 *FA 15C DY43531 *
V001B2A40 900ED00C 189F45E0 99765830 D1589110 06 *î } _É\rn¿ J¿j *
V001B2A50 A06C4780 90325830 D15C5030 D1081861 06 *+%_°î ¿ J*& J /*
AR+0015

15
AR+0015 MATCH FOUND AT 002CEA84
V002CEA80 ..... C9D2D8C2 C6C1404D 40404040 06 *....IKQBFA ( *
V002CEA90 40404040 40404040 40404040 40404040 06 * *
V002CEAA0 40404040 40404040 40404040 40404040 06 * *
V002CEAB0 40404040 40404040 40404040 40404040 06 * *

15 E
AR 0015 1I40I  READY

```

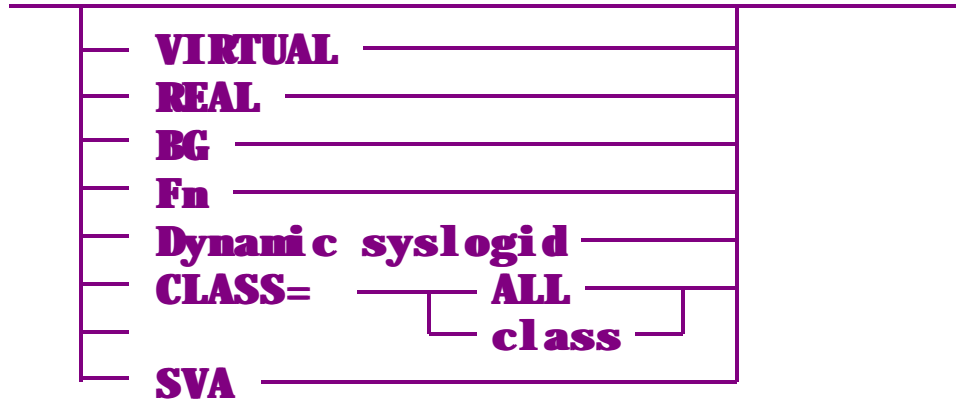
Be sure to terminate the LOCATE command by entering “15” plus any non-blank character, for instance “15 E”. The attention routine remains locked (blocked) until the command is terminated.

# PD/PSI Console Commands

## MAP

See “VSE/ESA System Control Statements” (SC33-6613)

### MAP



MAP VIRTUAL is the default.

## PD/PSI Console Commands

### Speaker Notes:

MAP							
AR	0015	SPACE	AREA	V- SIZE	GETVIS	V- ADDR	UNUSED NAME
AR	0015	SPACE	AREA	V- SIZE	GETVIS	V- ADDR	UNUSED NAME
AR	0015	S	SUP	704K		0	\$\$\$\$SUPX
AR	0015	S	SVA- 24	1832K	1752K	B0000	832K
AR	0015	0	BG V	1280K	4864K	500000	24576K
AR	0015	1	F1 V	1024K	2048K	500000	OK POWSTART
AR	0015	2	F2 V	2048K	28672K	500000	OK CICSICCF
		.					
		.					
		.					
AR	0015	B	FB V	256K	256K	500000	OK SECSERV
AR	0015	S	SVA- 31	7420K	6916K	2300000	
AR	0015		DYN- PA	OK			
AR	0015		DSPACE	5472K			
AR	0015		SYSTEM	1024K			
AR	0015		AVAIL	133088K			
AR	0015		TOTAL	259584K	<- - - - '		

MAP F2			
AR	0015	PARTITION: F2	SPACE-GETVIS. . . . . : (N/A)
AR	0015	SPACE. . . . : 2	ALLOC (VIRTUAL) . . . : 30720K ADDR: 500000
AR	0015	STATUS. . . : VIRTUAL	SIZE. . . . . : 2048K
AR	0015	POWER- JOB: CICSICCF	EXEC- SIZE. . . . . : 4K
AR	0015	JOBNUMBER: 612	GETVIS. . . . . : 28672K
AR	0015	JOBNAME. . : CICSICCF	EXEC- GETVIS. . . . : 30716K ADDR: 501000
AR	0015	PHASE. . . . : DFHSIP	
AR	0015		PFI X(BELOW) - LI MI T : 144K
AR	0015		- ACTUAL: 28K
AR	0015		PFI X(ABOVE) - LI MI T : OK
AR	0015		- ACTUAL: OK

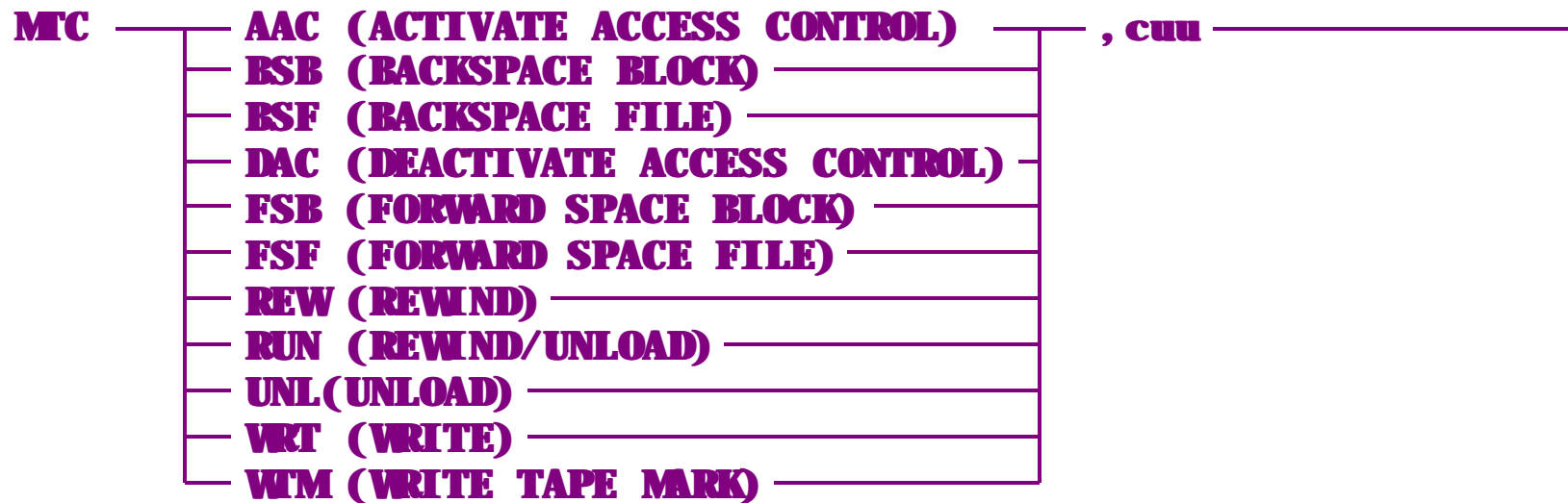


# PD/PSI Console Commands

## MTC (Magnetic Tape Command)

See “*VSE/ESA 2.6 System Control Statements*” (SC33-6613)

MTC is a Job Control Statement and an Attention Routine Command.



## PD/PSI Console Commands

### NOASSGN {OFF}

Instructs VSE not to issue a Tape **ASSIGN CCW** command when a shared tape drive is used. “**NOASSIGN OFF**” reinitializes normal “**ASSIGN / UNASSIGN**” processing. (See write-up on page 62.)

### NOAVR {OFF}

Turns off Automatic Volume Recognition (dasd and tape). “**NOAVR OFF**” turns it back on. (See write-up on page 62.)

### NOUNL {OFF}

Automatically converts Tape “Unload” commands to “Rewind”. Same as “**TAPE UNL=REW**”

### ONLINE cuu ,FORCE

Forces an update of Device Characteristics (issues **READ RDC CCW** command, x'64') even if device was not previously in “device down” status.

### READY cuu

Resets I/O status flag for this device, and generates a pseudo-interrupt.

## PD/PSI Console Commands

### PAUSE

See “VSE/ESA System Control Statements” (SC33-6613)

**PAUSE** [partition syslogid | luname | jobname] [,EOJ] ]

Pauses execution in the indicated partition at the next job control boundary (i.e. after currently executing phase / command completes) or at end-of-job, if “,EOJ” is specified.

### SUSPEND

**SUSPEND** [task id | partition syslogid]

Pauses execution for the indicated task / partition immediately. Effectively moves partition out of dispatch pool. If a partition syslog id is specified (e.g. “BG”, “F7”), the main task and all VSE sub-tasks are suspended.

### RESUME

**RESUME** [task id | partition syslogid]

Resumes execution for the indicated task / partition immediately. Effectively moves partition back into dispatch pool.

## PD/PSI Console Commands

### Speaker Notes:

The Attention Routine (AR) **DUMP** command to tape is a quick and effective way to obtain a snapshot of a batch partition (particularly CICS TS) without having to take a stand-alone dump and re-ipl the system. The format of the dump on tape is the same as a Stand-alone tape. By following the attached JCL, the supervisor and SVA will be dumped to a single tape file , and can be loaded into a VSEDUMP library using INFOANA or Interactive Interface 4.3.

This procedure does not work for problems with the VSE/ESA supervisor routines, since the INFOANA analyze routines (IJBXDEBUG, IJBXSDA, and IJBXCSMG) only work with stand-alone dumps, not with operator-initiated dumps.

However, the downside of using AR **DUMP** is that the system (including the batch and on-line partitions) continue to run while the dump is being taken, which may result in an inconsistent dump, where the higher addresses (dumped later) reflect a later point-in-time than the lower addresses (dumped earlier). To resolve this, use the **SUSPEND** and **RESUME** commands.

```
MTC REW,181
AR 0015 1I40I  READY
SUSPEND F2
AR 0015 1I40I  READY
DUMP SUP,181
AR 0015 1I51I  DUMP COMPLETE
AR 0015 1I40I  READY
MTC BSE,181
AR 0015 1I40I  READY
DUMP SVA,181
AR+0015 1I59D ENTER PHASE NAME, SVA24, GETVIS24, SVA31, GETVIS31 OR ALL
15 ALL
AR 0015 1I51I  DUMP COMPLETE
AR 0015 1I40I  READY
DUMP F2,181
AR 0015 1I51I  DUMP COMPLETE
AR 0015 1I40I  READY
RESUME F2
AR 0015 1I40I  READY
MTC REW,181
AR 0015 1I40I  READY
```

## PD/PSI Console Commands

### REIPL

See "*Hints and Tips for VSE*"

**REIPL [ CUU [ , NOPROMPT ] ]**

- Does exactly what it sounds like. Use with extreme care.
- In fact, you may want to use the STACK command to disable this command.

```
REIPL
AR+0015 1IxxD RE- IPL FROM CUU=120? REPLY YES OR NO
15 YES

***** LOGGING RESUMED AFTER VSE SYSTEM RE- IPL *****
BG 0000 0I04I IPLDEV=X' 120' , VOLSER=CTS120, CPUID=FFFF10009672
BG 0000 0J01I IPL=$IPLESA , JCL=$$JCL
BG 0000 $$ASSUPX, VSIZE=120M VIO=512K, VPOOL=64K, LOG
```

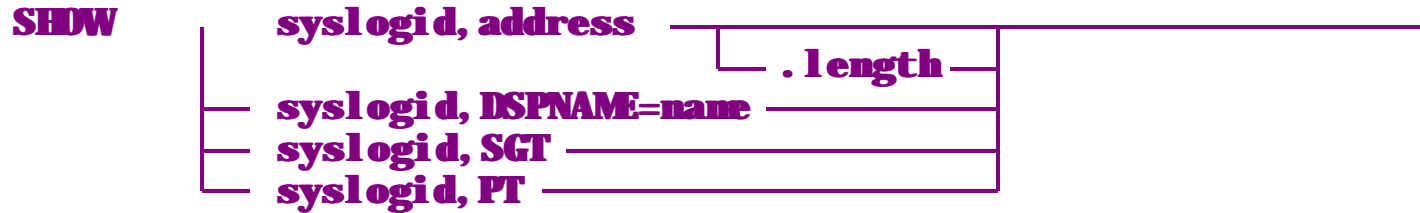
### PWROFF

- Initiates a hardware power off sequence. Designed for remote operations.
- Under VM, logs off the VSE virtual machine.
- Also a good candidate for the STACK command disabling.

# PD/PSI Console Commands

## SHOW

See “*VSE/ESA Supervisor Diagnosis Reference*” (LY33-9164)



- Displays up to 4K on the console.
- Memory will be paged in, if required.
- If length is not specified, displays 32 bytes (x'20').

```
SHOW F2,600000.3F
AR 0015  DATA  FOUND AT 00600000
V00600000 C4C6C8E2 C9D74040 07BD0000 00606B0A B6 *DFHSIP ] -, *
V00600010 00000000 00603F78 006000A8 00603280 B6 * - 1 - y - °*
V00600020 00601980 00000000 00608A18 008F1150 B6 * - ° - û &*
V00600030 006C2800 00603380 00601A24 00605F78 B6 * % - ° - - ^1*
AR 0015 1I40I  READY
```

## PD/PSI Console Commands

### SIR

See “*Hints and Tips for VSE*”

<b>SIR</b> (blank)	Full status display
<b>HELP</b>   ?	List available SIR commands
<b>SYS</b>   <b>SYSLEVEL</b>	Static status information
<b>RESET</b>	Reset dynamic counters
<b>SMF</b> [,VSE] = [ON   OFF   cuu ]	Subsystem Measurement Facility (When measured by the hardware, some SMF counters are inaccurate. “,VSE” counts from the VSE side.)
<b>MON</b> [= id   ON [ , NOSYM ]   OFF [ ,FAST   COUNTER   BOUND ]	Turbo Dispatcher statistics
<b>MIH</b> [= nnnnnn   ON   OFF ]	Missing Interrupt interval
<b>CHPID</b> [= id]	Channel paths defined to system
<b>CRWMSG</b> = ON   OFF	Channel Report Word information
<b>VENDOR</b>	Vendor product information
<b>VTAPEBUF</b> (=<nnnK nnM>)	Change VTAPE Buffer Size (128K ≤ nnn ≤ 16Meg)

## PD/PSI Console Commands

### Speaker Notes:

SIR	SIR		
SIR SYSLEVEL	AR 0015	CPUID VM = 0004941020640000	VSE = FF04941020640000
	AR 0015	VM-SYSTEM = VM/ESA (LPAR) 4.4.0	0301
	AR 0015	PROCESSOR = 2064-00	USERID = VSETEST4
	AR 0015	PROC-MDE = ESA (64-BIT)	IPL(140) 17:45:13 EDT 10/04/2003
	AR 0015	SYSTEM = VSE/ESA	2.7.0 GA 02/25/2003
	AR 0015	VSE/AF	6.7.0 DY46053 08/25/2003
	AR 0015	VSE/POWER	6.7.0 DY BASE 12/02/2002
	AR 0015	IPL-PROC = \$IPLESA	JCL-PROC = \$\$JCL
	AR 0015	SUPVR = \$\$\$SUPX	TURBO-DISPATCHER (40) ACTIVE
	AR 0015		HARDWARE COMPRESSION ENABLED
	AR 0015	SEC. MGR. = BASIC	SECURITY = ONLINE
	AR 0015	VIRTCPU = 0000:00:06.981	CP = 0000:00:01.426
	AR 0015	CPU-ADDR. = 0000(IPL) ACTIVE	
	AR 0015	ACTIVE = 0000:00:05.214	WAIT = 0000:58:02.502
	AR 0015	PARALLEL= 0000:00:02.630	SPIN = 0000:00:00.000
	AR 0015	CPU timings MEASUREMENT INTERVAL	0000:58:09.498
	AR 0015	TASKS ATT. = 00017	HIGH-MARK = 00017 MAX = 00184
	AR 0015	DYN. PARTS = 00000	HIGH-MARK = 00000 MAX = 00028
	AR 0015		
	AR 0015	COPY-BLKS = 00000	HIGH-MARK = 00032 MAX = 01500
	AR 0015	CHANQ USED= 00013	HIGH-MARK = 00020 MAX = 00255
	AR 0015	LBL. -SEGM = 00008	HIGH-MARK = 00008 MAX = 00717
	AR 0015	LOCKS EXT. = 0000000705	LOCKS INT. = 0000006604
	AR 0015	FAIL = 0000000018	FAIL = 0000000032
	AR 0015	LOCK I/O = 0000000000	LOCK WRITE= 0000000000



# PD/PSI Console Commands

## System Level:

System is at VSE/ESA 2.7.0, plus some maintenance (see previous page).

```
// EXEC PROC=SPLEVEL
BG 0000 * *****
BG 0000 *
BG 0000 *          VSE/ESA 2.7.0 GA   02/25/2003
BG 0000 *
BG 0000 * *****
BG 0000 * LICENSED MATERIALS - PROPERTY OF IBM
BG 0000 * 5686-066 AND OTHER MATERIALS (C) COPYRIGHT*
BG 0000 * IBM CORP. 2001 AND OTHER DATES
BG 0000 * ALL RIGHTS RESERVED.
BG 0000 * US GOVERNMENT USERS RESTRICTED RIGHTS -
BG 0000 * USE, DUPLICATION OR DISCLOSURE
BG 0000 * RESTRICTED BY
BG 0000 * GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
BG 0000 * *****
```

Updated for each FSU. Reflects level of last FSU installed. Never updated via PTF.  
Used by SIR as "VSE/ESA" level.

```
SHOW 2c0.40
AR 0015 DATA FOUND AT 000002C0
V000002C0 0027B000 0027B404 0000FB5C 0008E5E2 06 *.....*..VS* R000002C0
V000002D0 C561C1C6 40F5F6F8 F6F0F6F6 F0F640F7 06 *E/AF 568606606 7* R000002D0
V000002E0 F5C34040 F6F7F040 C4E8F4F6 F0F5F340 06 *5C 670 DY46053 * R000002E0
V000002F0 E5E2C54B C5E2C14B E2E4D7E7 40404040 06 *VSE. ESA. SUPX * R000002F0
AR 0015 1I40I READY
```

Updated by each VSE/ESA supervisor PTF.  
Used by SIR as "VSE/AF" level.

# PD/PSI Console Commands

Updated at release boundary  
Used by Vendor Products.

## SUBSID Macro (Supervisor Routine SGSSID):

```

119171+*-----
119172+*          SUBSYSTEM LIST STARTING WITH SUPERVISOR ENTRY
119173+*-----
041854 00000000
041858          119175+SSIDLST DC    0D' 0'          LIST AREA
041858 0000          119176+          DC    XL2' 00'          FLAGS
04185A 0000          119177+          DC    XL2' 00'          TASK-ID
          00004      119178+SSIDHDL EQU    *-SSIDLST          HEADER LENGTH
04185C 0000          119179+SSIDSUP DC    XL2' 00'          PARTITION-ID
04185E E2E4D740          119180+          DC    CL4' SUP '          NAME
041862 06          119181+          DC    AL1(6)          VERSION NUMBER
041863 07          119182+          DC    AL1(7)          RELEASE NUMBER
041864 00          119183+          DC    AL1(0)          MODIFICATION NUMBER
041865 06          119184+          DC    AL1(SSIDVLE-SSIDFB01) LENGTH OF VARIABLE PART
          0000A      119185+SSIDENL EQU    *-SSIDSUP          ENTRY LENGTH
041866 B8          119186+SSIDFB01 DC    BL1' 10111000'
041867 D0          119187+SSIDFB02 DC    BL1' 11010000'          FLAGBYTE 2
041868 60          119188+SSIDFB03 DC    BL1' 01100000'          FLAGBYTE 3
041869 00          119189+SSIDFB04 DC    BL1' 00000000'          FLAGBYTE 4 RESERV.
04186A 000F          119190+          DC    H' 15'          LENGTH OF LIB. CONCAT. CHAIN
          4186C      119191+SSIDVLE EQU    *          END OF VARIABLE PART
04186C 4B4B4B4B4B4B4B4B          119192+SSIDLST1 DC    (((SSTBEND- SSTABLE) /SSIDENL) +SSIDHDL) *36) C' . '
041CA4 FF          119193+SSIDLSTE DC    X' FF'          END OF LIST AREA
119195+*          RETURN CODES IN GR 15

```

```

LOCATE E2E4D74006
AR 0015  DATA  FOUND AT 0004185E
V00041850 ..... E2E4 06 *..... SU* R00041850
V00041860 D7400607 0006B9DA 2000000F 0001002C 06 *P ..... * R00041860
V00041870 0020E2C5 C3E20000 00000080 002200C0 06 *.. SECS. .... { * R00041870
V00041880 D7E6D940 06070000 00400024 00A0E5E3 06 *PWR ..... VT* R00041880
V00041890 C1D40000 00000010 00250090 C3C9C3F3 06 *AM ..... CIC3* R00041890
V000418A0 00000000 00100023 00B0C3C9 C3F40000 06 *..... CIC4. * R000418A0
V000418B0 00000001 005500B0 E2E2E740 00000000 06 *..... SSX .... * R000418B0
V000418C0 00200057 00B0C9C3 C3C60100 00004B4B 06 *..... ICCF..... * R000418C0
V000418D0 4B4B4B4B 4B4B4B4B 4B4B4B4B 4B4B4B4B 06 *..... * R000418D0

```

## PD/PSI Console Commands

### STACK | STACKP:

See “[\*Hints and Tips for VSE/ESA\*](#)”

**STACK** (blank) | **CLEAR** (Clears all active synonym definitions)  
**CLEAR=name** (Clears only specified synonym definition)  
**SHOW [ =name]** (Lists currently defined synonym(s))  
**name | command | command | ...** (may include special variables &0 .. &9  
**name** is the synonym name. A series of commands may be associated  
with this name by separating them with “|”.)

- Enables creation of a shortcut to a series of commands.
- Allows suppression or changing any VSE command.
- Abbreviates long VSE commands to just a few characters.
- Entire **STACK** command must not exceed 126 characters.
- **STACKP** creates a permanent synonym stack. (Writes to IPL device)
- A sample program in Hints and Tips shows how to define synonyms during IPL using SVC 30.

## PD/PSI Console Commands

### Speaker Notes:

```
STACK BG | R RDR,PAUSEBG
AR 0015 1I40I  READY
BG
AR 0015 1I40I  READY
AR 0015  R RDR, PAUSEBG
AR 0015 1C39I  COMMAND PASSED TO VSE/POWER
F1 0001 1R88I  OK
BG 0001 1Q47I  BG PAUSEBG 00023 FROM (SYSA) , TIME= 1:16:20
BG 0000 // JOB PAUSEBG
        DATE 03/16/1999, CLOCK 01/16/20
BG-0000 // PAUSE

STACK PAUSE | R RDR,PAUSE&0
AR 0015 1I40I  READY
PAUSE BG
AR 0015 1I40I  READY
AR 0015  R RDR, PAUSEBG
AR 0015 1C39I  COMMAND PASSED TO VSE/POWER
F1 0001 1R88I  OK
BG 0001 1Q47I  BG PAUSEBG 00261 FROM (SYSA) , TIME= 1:19:54
BG 0000 // JOB PAUSEBG
        DATE 03/16/1999, CLOCK 01/19/54
BG-0000 // PAUSE
```

# PD/PSI Console Commands

## STATUS

See "VSE/ESA System Control Statements" (SC33-6613) and "Hints and Tips for VSE/ESA"

```
STATUS
AR 0015 M12 CST      82 WAITING FOR I/O ON DEVICE=009
AR 0015 M20 AR       83 READY TO RUN
AR 0015 M21 BG       57 VSE/POWER WAITING FOR WORK
AR 0015 S3B - F1     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 M22 F1      82 WAITING FOR I/O, OR ECB POSTING      POWER MAIN TASK
AR 0015 S40 - F2     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 S41 - F2     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 S42 - F2     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 S43 - F2     82 WAITING FOR I/O, OR ECB POSTING
.
.
.
AR 0015 M2A F9       57 VSE/POWER WAITING FOR WORK
AR 0015 M2B FA       57 VSE/POWER WAITING FOR WORK
AR 0015 S39 - FB     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 S3A - FB     82 WAITING FOR I/O, OR ECB POSTING
AR 0015 M2C FB      82 WAITING FOR I/O, OR ECB POSTING      SECURITY SERVER
AR 0015 1I40I READY

STATUS F2
AR 0015 S40-F2 EVA10MST 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004980C PCB=0004A200 TCB=002C461C TIB=002C45A0 SAV=00405260
AR 0015 S41-F2 DFHEVID2 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004980C PCB=0004A200 TCB=002C48EC TIB=002C4870 SAV=00292400
AR 0015 S42-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004980C PCB=0004A200 TCB=002C4BBC TIB=002C4B40 SAV=00292480
AR 0015 S43-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004980C PCB=0004A200 TCB=002E607C TIB=002E6000 SAV=00292500
AR 0015 S44-F2 DFHSKTSK 82 WAITING FOR I/O, OR ECB POSTING
.
.
.
```

## PD/PSI Console Commands

STATUS 240

```
AR 0015 SCHIB DEV INT-PARM ISC FLG LP PNO LPU PI MBI PO PA CHPID0-3 CHPID4-7
AR 0015 000A 0240 00003418 3 83 E0 00 20 F0 0000 F0 E0 24272A28 FFFFFFFF
AR 0015 CHPIDS NOT ACCESSIBLE: 28
```

STATUS 480, ALL

```
AR 0015 SCHIB DEV INT-PARM ISC FLG LP PNO LPU PI MBI PO PA CHPID0-3
AR 0015 0010 0480 00003710 3 83 80 00 80 80 0000 80 80 0AFFFFFFF

AR 0015 KEY SLCC FPIAUZEN FCTL ACTL SCTL CCWADDR DS CS CNT
AR 0015 0 0 00 40 0000 07 00027358 0C 00 000C

AR 0015 REQUEST IS STARTED DEVICE IS BUSY
AR 0015 PUB=00003710 PUBX=0007A228 PUB2=00072288 POWN=00003BBC
AR 0015 VCTE=000712FA POWNX=00237BC4
```

**SCHIB** Subchannel number of the device that has been inspected.

**DEV** Device number

**INT-PARM** Interrupt parameter

**ISC** Interruption subclass code

**FLG** Flag field

**LP** Logical Path Mask

**PNO** Path Not Operational Mask

**LPU** Last Path Used Mask

**PI** Path Installed Mask

**MBI** Measurement Block Index

**PO** Path Operational Mask

**PA 5** Path Available Mask

**CHPID0-3** Channel Path Identifiers 0 through 3

**CHPID4-7** Channel Path Identifiers 4 through 7

**KEY** Storage protection key used at SSCH time

**SLCC** Progress of I/O operation (cc = deferred condition code)

**FPIAUZEN** Control bits

**FCTL** Function control information (40 = start, 20 = halt, 10 = clear)

**ACTL** Activity control information (04 = start pending, 02 = halt pending, 01 = clear pending, byte 2: 80 = subchannel active, 40 = device active)

**SCTL** Status control information (10 = Alert status, 04 = primary I/O interrupt status, 02 = secondary interrupt status, 01 = status pending)

**CCW-ADDR** Address+8 of last CCW that was executed

**DS** Device Status information

**CS** Channel status information

**CNT** Residual count

## PD/PSI Console Commands

### TAPE:

See "[Hints and Tips for VSE/ESA](#)"

```
TAPE — UNL={UNL | REV} —————  
      |  
      |— DSPLY={VOL | JNM} ————|  
      |  
      |— WTM={SYNC | NOSYNC} ———|
```

**UNL**      Optionally converts rewind-unload tape commands to a simple rewind.  
            See also "NOUNL" command.

**DSPLY**   Changes the information normally displayed on the Load-Display (LED) of 3480, 3490, and  
            3590 tape cartridge devices.  
            "JNM" will cause the volume and job name to alternately flash each 2 seconds.

**WTM**      Synchronize Write Tape Mark

## PD/PSI Console Commands

This page  
left  
intentionally  
blank



# PD/PSI Console Commands

## TERSE:

See “*VSE/ESA 2.7 Release Guide*”

```
// EXEC TERSE,PARM='DD:lib.sublib(member.type) DD: lib.sublib(member.type)'
```

- Cross-Platform compression utility. (Compatible with VM/ESA and OS/390)
- Only works on VSE/ESA library members (not DUMP or PHASE)
- Automatically detects if file should be compressed or uncompressed. (can be controlled via switches.)
- Uses temporary space in same library as input file
- Does not use Hardware Compression (Nagy algorithm).

## TIME:

See “*Hints and Tips for VSE/ESA*”

```
TIME [ DATE=MM/DD/YYYY, CLOCK=HH/MM/SS ]  
[ ZONE = EAST | WEST | zone- id/hh/mm  
[ ZONE = VM
```

- Allows operator to display or alter the current time or time zone.
- Extreme care must be exercised in using this command. Setting the time back may result in system mal-function, and render job accounting and/or journaling data invalid.

# PD/PSI Console Commands

## Speaker Notes:

```
// JOB TERSE A LIBRARY MEMBER  
// EXEC TERSE, PARM=' DD: IJSYSRS. MLIB(INPUT. COPY) DD: IJSYSRS. MLIB(OUTPUT.  
T. TERSED) '  
/*  
/&
```

### LIBR List Directory Member:

<b>INPUT</b>	<b>COPY</b>	<b>CREATION DATE</b>	<b>:</b>	<b>2003-10-04</b>	<b>17:46</b>
		<b>LAST UPDATE</b>	<b>:</b>	<b>- -</b>	
		<b>NUMBER OF RECORDS</b>	<b>:</b>	<b>349</b>	
		<b>LOGICAL RECORD SIZE:</b>		<b>80</b>	
		<b>LIBRARY BLOCKS USED:</b>		<b>11</b>	
<b>OUTPUT</b>	<b>TERSED</b>	<b>CREATION DATE</b>	<b>:</b>	<b>2003-10-04</b>	<b>18:00</b>
		<b>LAST UPDATE</b>	<b>:</b>	<b>- -</b>	
		<b>NUMBER OF RECORDS</b>	<b>:</b>	<b>64</b>	
		<b>LOGICAL RECORD SIZE:</b>		<b>80</b>	
		<b>LIBRARY BLOCKS USED:</b>		<b>6</b>	

## PD/PSI Console Commands

### **INTERACTIVE TRACE:**

See "VSE/ESA Diagnosis Tools" (SC33-6614)

**// EXEC phase, TRACE**

- ",TRACE" parameter activates Interactive Trace Program
- Tracing is only within this user partition
- Control is passed to console operator at beginning of phase execution.
- Uses hardware PER feature, so cannot be used with SDAIDs.

### **TRACE Types:**

- **BRANCH** (ADDRESS={address | address.length | address1:address2})
- **INSTRUCTION** (ADDRESS={address | address.length | address1:address2})
- **STORAGE** (ADDRESS={address | address.length | address1:address2})
- **ABEND**
- **QUERY**
- **DISPLAY** {Address | Address.length | Address1:Address2}
- **ALTER**           address   **DATA** = data  
                          **GN DATA** = data
- **GO** (address)   (**OUTPUT=SYSLST**)  
                          (**OPTION** = {**DUMP** | **PARTDUMP** | **NODUMP** })
- **TRACE END** (nn)

# PD/PSI Console Commands

## Speaker Notes:

- **BRanch (ADDRESS={address | address.length | address1:address2})**  
If target of branch is within specified address range.
- **Instruction (ADDRESS={address | address.length | address1:address2})**  
If first byte of instruction within specified address range
- **STorage (ADDRESS={address | address.length | address1:address2})**  
Attempts to change storage within specified address range. Even if value stored is the same as the original value. Monitoring does not detect data changed by a channel program or by system control programs.
- **ABend**  
The ABEND "abnormal end" trace allows interactive debugging if a user program terminates abnormally. In case of an ABEND, the termination routines display the cancellation message on the screen and transfer control to the console operator. The operator can inspect storage data or register contents to determine the cause of the cancellation. It is, however, not possible to change the program status and return to normal operation via the GO command with a branch address. The task termination is already in progress at that time. It is, however, possible to modify the dump option to DUMP, PARTDUMP, or NODUMP

In case of BRanch, Instruction, and STorage traces, the traced instruction has already been executed.

- **GO (address) (OUTPUT=SYSLST) |(OPTION = {Dump | Partdump | Nodump})**  
If the address parameter is omitted, the program continues processing with the next sequential instruction. The parameter OUTPUT=SYSLST switches the tracing mode from interactive tracing to batch tracing. The parameter OPTION= modifies the temporary dump options. In case of an abnormal termination, the dump routines will print either a full dump (DUMP), a partition dump (PARTDUMP), or no dump at all (NODUMP). If the parameter OPTION is omitted, the dump options remain unchanged.  
A replid without trace command defaults to an unqualified GO command. That is, instruction tracing will resume with the next instruction.
- **TRACE END (nn)** Terminates indicated (or all) trace(s).

## PD/PSI Console Commands

Example:

```
// JOB ABC
// EXEC PROG1, TRACE
/ &
```

```
BG 0000 4I01I TRACE STARTED FOR PROGRAM PROG1
BG-0000 00400078 BALR 05C0 CC 0

0 QUERY
BG 0000 001 TRACE INST ADDRESS=00400000: 004AFFFF
BG-0000 002 TRACE ABEND

0 TRACE END 1
BG-0000 4I09D SPECIFIED TRACE ENDED

0 TRACE INST ADDRESS=403BA0.70
BG-0000 003 TRACE INST ADDRESS=00403BA0: 00403COF
0 trace stor address=4002ad
BG-0000 004 TRACE STOR ADDRESS=004002AD: 004002AD

0 TRACE INST ADDRESS=4017CC:4017FF
BG-0000 005 TRACE INST ADDRESS=004017CC: 004017FF

0 QUERY
BG 0000 002 TRACE ABEND
BG 0000 003 TRACE INST ADDRESS=00403BA0: 00403COF
BG 0000 004 TRACE STOR ADDRESS=004002AD: 004002AD
BG-0000 005 TRACE INST ADDRESS=004017CC: 004017FF

0
BG-0000 0040007A B 47103024 -> 0040009E CC 0

0
BG-0000 0040009E NOPR 0700 CC 0

63 GO OUTPUT=SYSLST
BG 0000 4I20I TRACING TERMINATED
```

## PD/PSI Console Commands

```
***** START OF BATCH TRACE *****
0000 004000A0 BAL 4510303E -> 004000B8 CC 0
0000 004000B8 LR 1801 CC 0
0000 004000BA SVC 0A26 CC 0
0063 004001A4 LA 41603313 = 0040038D CC 0
0063 004001A8 STCM BE67359F >> 00400619 CC 0
0063 004001AC LA 41600020 = 00000020 CC 0
0063 004001B0 STC 426035A5 >> 0040061F CC 0
0063 004001B4 L 58103606 00400680 CC 0
0063 004001B8 SVC 0A00 CC 3
0063 004001BA L 58103606 00400680 CC 0
0063 004001BE TM 91801002 004005DE CC 0

5 DISPLAY PSW
F5-0005 PSW = 470D0000 00400EE6

5 DISPLAY GR
F5 0005 GPR 0 = 00000000 00400648 00003110 00400EF4
F5 0005 GPR 4 = 00403518 004001AE 004010E0 004000E0
F5 0005 GPR 8 = 00407000 004020E0 00000590 004030E0
F5-0005 GPR 12 = 00000006 00407894 804017FC 90000004

5 ALTER GC DATA=00000007
F5-0005 GPR 12 = 00000007

5 DISPLAY 403017.20
F5 0005 00403010 00E1D8E0 00E1474E 00006FBC 00000644 *.....*
F5 0005 00403020 00009DA0 00E263C8 00E18598 00FC4540 *.....*
F5-0005 00403030 00004930 000094E8 00E623F8 00E26414 *.....*

5 DISPLAY 403017
F5-0005 00403010 00E1D8E0 00E1474E 00006FBC 00000644 *.....*

5 ALTER 403017 DATA=FEFEFE
F5-0005 00403010 00E1D8E0 00E147FE FEFE6FBC 00000644 *.....*
010 00E1D8E0 00E147FE FEFE6FBC 00000644 *.....*
```

## PD/PSI Console Commands

### VOLUME (tape)

See “[\*Hints and Tips for VSE/ESA\*](#)”

- Enhancement to traditional VOLUME command, to show additional information for tape units.
- Output will be similar to following:

<b>volume</b>	<b>tape</b>							
<b>AR</b>	<b>0015</b>	<b>CUU</b>	<b>CODE</b>	<b>DEVICE-ID</b>	<b>VOLID</b>	<b>USE/STATUS</b>	<b>- INFORMATION</b>	<b>CAPACITY</b>
<b>AR</b>	<b>0015</b>	<b>480</b>	<b>5400</b>	<b>3490-40</b>	<b>TAP634</b>	<b>BG</b>	<b>BUFD</b>	<b>22356</b>
<b>AR</b>	<b>0015</b>	<b>481</b>	<b>5400</b>	<b>3490-40</b>	<b>*none*</b>	<b>F4</b>	<b>SYNC</b>	<b>0</b>
<b>AR</b>	<b>0015</b>	<b>482</b>	<b>5400</b>	<b>3490-40</b>	<b>ISMNE</b>	<b>BG</b>	<b>2XF SYNC</b>	<b>1</b>
<b>AR</b>	<b>0015</b>	<b>483</b>	<b>5400</b>	<b>3490-40</b>	<b>TAP634</b>	<b>BG</b>	<b>NOT READY</b>	
<b>AR</b>	<b>0015</b>	<b>484</b>	<b>5400</b>	<b>TAPE</b>		<b>SHARED</b>	<b>NOT OPER.</b>	

**CUU** device number as known to VSE  
**CODE** device type code and mode setting currently active for this device  
**DEVICE-ID** the device-type and model  
**VOLID** VOL1 label (if any) of the media currently or last mounted on tape drive  
**USE/STATUS** contains information about owner of the tape drive (if an owner exists)  
**INFORMATION** media format mode (2XF vs XF) and whether operating as SYNC or BUFD  
**CAPACITY** last block (record) read from, or written to the device. Zero indicates at load point.  
This value increments, allowing you to monitor tape job run progress.