

VAX/VMS Internals

VMS V4.4
Supplementary Update Material

Prepared by Educational Services
of
Digital Equipment Corporation

Copyright © 1986 by Digital Equipment Corporation
All Rights Reserved

The reproduction of this material, in part or whole, is strictly prohibited. For copy information, contact the Educational Services Department, Digital Equipment Corporation, Bedford, Massachusetts 01730.

Printed in U.S.A.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may not be used or copied except in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

The manuscript for this book was created using **DIGITAL Standard Runoff**. Book production was done by **Educational Services Development and Publishing in Nashua, NH**.

The following are trademarks of Digital Equipment Corporation:

digital™

DATATRIEVE

DEC

DECmate

DECnet

DECset

DECsystem-10

DECSYSTEM-20

DECtape

DECUS

DECwriter

DIBOL

MASSBUS

PDP

P/OS

Professional

Rainbow

RSTS

RSX

UNIBUS

VAX

VMS

VT

Work Processor

VMS V4.4 SUPPLEMENTARY UPDATE MATERIAL

INTRODUCTION

Since VAX/VMS V4.0 was released, changes have been made to the System Dump Analyzer (SDA) and the VAX/VMS documentation has been enhanced. This update module gives an overview of the new System Dump Analyzer (SDA) commands and summarizes the changes to the VAX/VMS documentation.

OBJECTIVES

To perform your job effectively on a VAX/VMS V4.4 system, you should be able to:

- Recognize the changes in the VAX/VMS documentation set
- List and use the new System Dump Analyzer (SDA) commands

CHANGES IN VAX/VMS DOCUMENTATION

Table 1 Documentation Changes for VAX/VMS V4.4

V4.0 or V4.2 Manual	Change	New Manual(s)
VAX/VMS DCL Dictionary	Split into 2 manuals	VAX/VMS DCL Concepts Manual (contains Part I, Using the Using the Command Language) VAX/VMS DCL Dictionary (contains Part II, Command Descriptions)
User's Guide to EVE	Combined with another manual	Guide to Text Processing on VAX/VMS
VAXTPU EDT Keypad Emulator Quick Reference Guide	Combined with another manual	Guide to Text Processing on VAX/VMS
Guide to VAX/VMS System Management and Daily Operations	Name change	VAX/VMS System Manager's Reference Manual
VAX/VMS Symbolic Debugger Reference Manual	Name change	VAX/VMS Debugger Reference Manual
Guide to Networking on VAX/VMS	Name change	VAX/VMS Networking Manual

OVERVIEW OF THE NEW SYSTEM DUMP ANALYZER (SDA) COMMANDS

Since VAX/VMS V4.0 was released, the following new SDA commands have been added:

- ATTACH

- SPAWN

Also, new qualifiers are available for the EVALUATE, EXAMINE, and SEARCH commands as follows:

- EVALUATE /PSL

- EVALUATE /PTE

- EVALUATE /SYMBOLS

- EXAMINE /NOSUPPRESS

- EXAMINE /PTE

- SEARCH /LENGTH=length_specifier

- SEARCH /STEPS=step_factor

EXAMPLES AND ADDITIONAL INFORMATION

The ATTACH and SPAWN Commands

The ATTACH command allows you to switch control of your terminal to another process in your job. The /PARENT qualifier allows you to switch control of your terminal to the parent process of the current process.

The SPAWN command creates a subprocess from the current process. The context is copied from the current process to the spawned process.

```
$ ANALYZE/SYSTEM
```

```
VAX/VMS System analyzer
```

```
SDA> SPAWN
```

```
$ ATTACH MARSH
```

```
SDA> ATTACH MARSH_1
```

```
%DCL-S-RETURNED, control returned to process MARSH_1
```

```
$ log
```

```
Process MARSH_1 logged out at 19-JAN-1986 14:53:54.61
```

```
SDA>
```

Example 1 Using the ATTACH and SPAWN Commands

The EVALUATE/PSL Command

The EVALUATE/PSL command evaluates the specified longword, displaying its contents in the format of a processor status longword.

In this example, three steps are shown:

Step 1 Find and format the current PCB

Step 2 From step 1, find and format the PHD

Step 3 From step 2, find and format the PSL

Step 1

```
SDA> EXAMINE SCH$GL_CURPCB
SCH$GL_CURPCB: 80110A60 ``...''
SDA> READ SYS$SYSTEM:SYSDEF.STB
SDA> FORMAT @SCH$GL_CURPCB
```

80110A60	PCB\$L_SQFL	8000201C	
80110A64	PCB\$L_SQBL	8000201C	
80110A68	PCB\$W_SIZE	0120	
80110A6A	PCB\$B_TYPE	0C	
.	.	.	.
.	.	.	.
80110AC0	PCB\$L_PID	000E0003	
80110AC4	PCB\$L_EPID	000000E3	
80110AC8	PCB\$L_EOWNER	00000000	
80110ACC	PCB\$L_PHD	8019AA00	<---- PHD Address
80110AD0	PCB\$t_LNAME	"MARSH"	
.	.	.	.
.	.	.	.
80110B78	PCB\$L_WAITIME	002173DE	
80110B7C	PCB\$L_PMB	00000000	
	PCB\$C_LENGTH		

Example 2 Using the EVALUATE/PSL Command
(Sheet 1 of 2)

The EVALUATE/PSL Command

Step 2

```
SDA> FORMAT`G19AA00
%SDA-E-NOSYMBOLS, no "PAGED" symbols found to format this block
```

Note: The PHD has no TYPE field. The /TYPE qualifier must be added to the SDA> FORMAT command

```
SDA> FORMAT/TYPE=PHD G19AA00
```

8019AA00	PHD\$Q_PRIVMSK	00108001	
8019AA04		00000000	
8019AA08	PHD\$W_WSLIST	0063	
8019AA0A	PHD\$W_WSAUTH	0262	
8019AA0C	PHD\$W_WSLOCK	0071	
8019AA0E	PHD\$W_WSDYN	007D	
8019AA10	PHD\$W_WSNEXT	00DA	
8019AA12	PHD\$W_WSLAST	0382	
.	.	.	.
.	.	.	.
8019AAB4	PHD\$L_R11	0C0C0007	
8019AAB8	PHD\$L_R12	7FFE9DC4	
8019AABC	PHD\$L_R13	7FFE7DE4	
8019AAC0	PHD\$L_PC	8011400A	
8019AAC4	PHD\$L_PSL	00420008	<----- Process PSL
8019AAC8	PHD\$L_POBR	8019C400	
8019AACC	PHD\$L_POLRASTL	0400079E	
	PHD\$B_ASTLVL		
8019AAD0	PHD\$L_P1BR	7F9A4400	
.	.	.	.
.	.	.	.
8019AB7C	PHD\$L_WSL	00000000	

Step 3

```
SDA> EVALUATE/PSL @G19AAC4
```

```

CMP TP FPD IS CURMOD PRVMOD IPL DV FU IV T N Z V C
0 0 0 0 KERN EXEC 02 0 0 0 0 1 0 0 0
```

Example 2 Using the EVALUATE/PSL Command
(Sheet 2 of 2)

The EVALUATE/SYMBOLS Command

The EVALUATE/SYMBOLS command specifies that all symbols that are known to be equal to the evaluated expression are to be displayed.

```
SDA> EVALUATE SCH$GL_CURPCB
Hex = 800021F8   Decimal = -2147474952   SCH$GL_CURPCB
SDA> EVALUATE/SYMBOLS G21F8
Hex = 800021F8   Decimal = -2147474952   SCH$GL_CURPCB
SDA> EVALUATE/SYMBOLS G
Hex = 80000000   Decimal = -2147483648
                                           G
                                           PCB$M_EPID_WILD
                                           PRV$M_SHARE
                                           PSL$M_CM
                                           VASM_SYSTEM

SDA> EVALUATE/SYMBOLS G14
Hex = 80000014   Decimal = -2147483628   SYS$CALL_HANDL+004
SDA> EVALUATE/SYMBOLS G200
Hex = 80000200   Decimal = -2147483136   SYS$CALL_HANDL+1F0
```

Example 4 Using the EVALUATE/SYMBOLS Command

The EXAMINE/NOSUPPRESS Command

The EXAMINE/NOSUPPRESS command inhibits the suppression of zeros when displaying memory with one of the following qualifiers: /ALL, /P0, /P1, /SYSTEM.

SDA> EXAMINE/P0

Process region memory

Virtual locations 00000000 through 000001FF are not in physical memory

0130011A	0120011B	0130011E	0110011F0... ..0.	00000200
01200107	02300510	04310216	04210218	..!...1...0... .	00000210
01100103	01100104	01200105	01200106	00000220
44412107	01100100	01100101	01100102!AD	00000230
4B202020	20444121	44412106	42582321	!#XB.!AD!AD K	00000240
00524553	55525055	53434558	454E5245	ERNEXECSUPRUSER.	00000250
53492044	50462050	5420504D	435F2138	8!_CMP TP FPD IS	00000260
4920444F	4D565250	20444F4D	52554320	CURMOD PRVMOD I	00000270
204E2054	20564920	55462056	44204C50	PL DV FU IV T N	00000280
00000000	0000005F	212F2143	2056205A	Z V C!//!	00000290

Zeros suppressed from 000002A0 through 0000040F

80110AE4	00000000	00108001	80142070	p	d... 00000410
00000450	00000450	00000450	00000450	P...P...P...P...	00000420

Example 5 Using the EVALUATE/P0 Command

VMS V4.4 SUPPLEMENTARY UPDATE MATERIAL

SDA> EXAMINE/P0/NOSUPPRESS

Process region memory

 Virtual locations 00000000 through 000001FF are not in physical memory

0130011A	0120011B	0130011E	0110011F0... ..0.	00000200
01200107	02300510	04310216	04210218	..!...1...0... .	00000210
01100103	01100104	01200105	01200106	00000220
44412107	01100100	01100101	01100102!AD	00000230
4B202020	20444121	44412106	42582321	!#XB.!AD!AD K	00000240
00524553	55525055	53434558	454E5245	ERNEXECSUPRUSER.	00000250
53492044	50462050	5420504D	435F2138	8!_CMP TP FPD IS	00000260
4920444F	4D565250	20444F4D	52554320	CURMOD PRVMOD I	00000270
204E2054	20564920	55462056	44204C50	PL DV FU IV T N	00000280
00000000	0000005F	212F2143	2056205A	Z V C!/_.....	00000290
00000000	00000000	00000000	00000000	000002A0
00000000	00000000	00000000	00000000	000002B0
.
.
.
00000450	00000450	00000450	00000450	P...P...P...P...	00000420

Example 6 Using the EVALUATE/P0/NOSUPPRESS Command

The EXAMINE/PTE Command

The EXAMINE/PTE command interprets and displays the specified longword as a Page Table Entry (PTE). The individual fields of the PTE are separated and an overall description of the PTE type is provided.

```
SDA> SHOW PAGE_TABLE G:G600
```

```
System page table
```

```
-----
```

ADDRESS	SVAPTE	PTE	TYPE	PROT	BITS
80000000	801EA200	F8000A34	VALID	UR	K
80000200	801EA204	F8000A35	VALID	UR	K
80000400	801EA208	F8000A36	VALID	UR	K

```
SDA> EXAMINE/PTE G1EA200
```

```

|31          28|27          24|23          20|19          16|15
|-----|-----|-----|-----|-----|-----|-----|-----|
|1 | 1 1 1 1 | 0 |--| 0 0 |--| 0|                000A34
+-----+-----+-----+-----+-----+-----+-----+-----+
Vld Prot= UR  M   Own=K   W                Page Frame Number

```

Page is Active and Valid

```
SDA> EXAMINE/PTE G1EA204
```

```

|31          28|27          24|23          20|19          16|15
|-----|-----|-----|-----|-----|-----|-----|-----|
|1 | 1 1 1 1 | 0 |--| 0 0 |--| 0|                000A35
+-----+-----+-----+-----+-----+-----+-----+-----+
Vld Prot= UR  M   Own=K   W                Page Frame Number

```

Page is Active and Valid

Example 7 Using The EXAMINE/PTE Command

The SEARCH/LENGTH Command

SEARCH

/LENGTH=length_specifier

This qualifier specifies the size of the expression value to be used for successful matching during searches of memory. The possible values of this qualifier are:

LONGWORD

Specifies that the expression to be searched for is four bytes in length. This is the default value.

WORD

Specifies that the expression to be searched for is two bytes in length.

BYTE

Specifies that the expression to be searched for is one byte in length.

The SEARCH/STEPS Command

SEARCH

/STEPS=step_factor

This qualifier controls the granularity of searching through the specified memory range. As each comparison of memory occurs, the value of this qualifier determines what the next memory location to be searched will be. The possible stepfactors are:

QUADWORD

Specifies a stepfactor of eight bytes.

LONGWORD

Specifies a stepfactor of four bytes. This is the default value for this qualifier.

WORD

Specifies a stepfactor of two bytes.

BYTE

Specifies a stepfactor of one byte.

Other Comments on SDA

Note that the COPY command releases the dump pages in the paging file so that they are available for system paging. Note that once the COPY command has released the dump pages for paging use, the dump information in these pages may be lost. Subsequent dump analysis should be carried out on the copy of the dump file that was specified in the COPY command.

The SET PROCESS and SHOW PROCESS commands can now include quoted strings in the process name in addition to the previous capital letters, numbers, dollar sign, and underscore.

The SHOW CRASH command register list now includes the system identification register, as shown in Example 8.

VMS V4.4 SUPPLEMENTARY UPDATE MATERIAL

SDA> SHOW CRASH

System crash information

Time of system crash: 19-JAN-1986 14:52:45.45

Version of system: VAX/VMS VERSION V4.4

Process currently executing: MARSH

Current image file: DUA0:[SYS0.] [SYSEXEC]SDA.EXE;1

Current IPL: 0 (decimal)

General registers:

R0 = 00000000	R1 = 00000000	R2 = 00000000	R3 = 00000000
R4 = 00000000	R5 = 00000000	R6 = 00000000	R7 = 00000000
R8 = 00000000	R9 = 00000000	R10 = 00000000	R11 = 00000000
AP = 00000000	FP = 00000000	SP = 00000000	PC = 00000000
PSL = 00000000			

Processor registers:

MicroVAX I

POBR = 00000000	SBR = 00000000	ASTLVL = 00000000
POLR = 00000000	SLR = 00000000	SISR = 00000000
P1BR = 00000000	PCBB = 00000000	ICCS = 00000000
P1LR = 00000000	SCBB = 00000000	SID = 07000401
TODR = 00000000		
ISP = 00000000		
KSP = 00000000		
ESP = 00000000		
SSP = 00000000		
USP = 00000000		

Example 8 Using the SHOW CRASH Command