OS/32 AIDS
User Guide

OS/32 Version 8.1 or higher

48-087 F00 R00

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include all copyright notices, trademarks, or other legends or credits of Concurrent Computer Corporation and/or its suppliers. Title to and ownership of the described software and any copies thereof shall remain in Concurrent Computer Corporation and/or its suppliers.

The licensed program described herein may contain certain encryptions or other devices which may prevent or detect unauthorized use of the Licensed Software. Temporary use permitted by the terms of the License Agreement may require assistance from Concurrent Computer Corporation.

Concurrent Computer Corporation assumes no responsibility for the use or reliability of this software if used on equipment that is not supplied by Concurrent Computer Corporation.

Concurrent Computer Corporation, 2 Crescent Place

Oceanport, New Jersey 07757

Printed in the United States of America

TABLE OF CONTENTS

PREFACE v

CHAPTERS

1 OS/32 AIDS

1.1 INTRODUCTION 1-1

1.2 BASIC OPERATIONS 1-1

1.3 REQUIREMENTS 1-1

1.4 FEATURES 1-2

1.5 MODES OF USE 1-2

1.6 DIRECTIVES 1-3

1.7 ADDRESS CONVENTIONS 1-3

1.8 LOGICAL UNIT (LU) ASSIGNMENTS 1-6

2 OS/32 AIDS DATA FORMAT MODES

2.1 INTRODUCTION 2-1

2.2 HALFWORD HEXADECIMAL FORMAT (X) 2-1

2.3 FULLWORD HEXADECIMAL FORMAT (Y) 2-1

2.4 HALFWORD DECIMAL FORMAT (H) 2-2

2.5 FULLWORD DECIMAL FORMAT (F) 2-2

2.6 CHARACTER FORMAT (C) 2-2

2.7 SINGLE PRECISION FLOATING POINT FORMAT (E) 2-2

2.8 DOUBLE PRECISION FLOATING POINT FORMAT (D) 2-3

2.9 DISASSEMBLY FORMAT (A) 2-4

CHAPTERS (Continued)

3 OS/32 AIDS COMMANDS

3.1	INTRODUCTION	3-1
3.2	ADD COMMAND	3-2
3.3	BATCH COMMAND	3-4
3.4	BIAS COMMAND	3-5
3.5	CONVERT COMMAND	3-6
3.6	DUMP COMMAND	3-9
3.7	END COMMAND	3-12
3.8	EXECUTE COMMAND	3-13
3.9	GO COMMAND	3-15
3.10	INSERT COMMAND	3-17
3.11	JUMP COMMAND	3-24
3.12	LOG COMMAND	3-26
3.13	LOGICAL UNIT (LU) COMMAND	3-27
3.14	NEXT COMMAND	3-28
3.15	NO BATCH COMMAND	3-29
3.16	NO LOG COMMAND	3-30
3.17	OPEN COMMAND	3-31
3.18	OPEN NEXT COMMAND	3-34
3.19	OPEN PREVIOUS COMMAND	3-36
3.20	PAUSE COMMAND	3-38
3.21	REPLACE COMMAND	3-39
3.22	SUBTRACT COMMAND	3-42
3.23	TYPE COMMAND	3-44
3.24	WHERE COMMAND	3-46
3.25	ZAP COMMAND	3-47

CHAPTERS (Continued)

4 OS/32 AIDS OPERATING INSTRUCTIONS

4.1	OS/32 AIDS OPERATION	4-1
4.2	START PROCEDURES	4-1
4.2.1	Building AIDS into a User Task (U-Task)	4-2
4.2.2	Building AIDS as a Shared Partial Segment	4-2
4.2.3	Using the START Command	4-4
4.3	ERROR MESSAGES	4-5

APPENDIXES

A	OS/32 AIDS COMMAND SUMMARY	A-1
B	SAMPLE DUMP FORMATS	B-1
C	SUPPLEMENT TO THE OS/32 AIDS USER GUIDE FOR CODE COMPATIBLE MACHINE (CCM) SUPPORT	
C.1	INTRODUCTION	C-1
C.2	GENERAL FEATURES	C-1
C.3	CODE COMPATIBLE MACHINE (CMM) SUPPORT FEATURES	C-1
C.3.1	Boundary Alignments	C-2
C.3.2	Breakpoint Sentinel	C-2
C.3.3	Fault Traps	C-2
C.3.4	User Task Status Word (TSW) Trap Handling	C-2
C.3.5	Data Format Options	C-2
C.3.5.1	Character Mode (C, CA, CE)	C-3
C.3.5.2	Disassembly Format (A, AP, AC)	C-3
C.3.6	Register Address Conventions	C-4
C.3.7	General Directive Guidelines	C-4
C.3.8	OS/32 AIDS Commands	C-5
C.3.8.1	CONVERT Command	C-6
C.3.8.2	DUMP Command	C-9
C.3.8.3	END Command	C-13
C.3.8.4	EXECUTE Command	C-14
C.3.8.5	GO Command	C-17
C.3.8.6	JUMP Command	C-19
C.3.8.7	OPEN Command	C-22
C.3.8.8	OPEN NEXT Command	C-26
C.3.8.9	OPEN PREVIOUS Command	C-29
C.3.8.10	REPLACE Command	C-32
C.3.9	Limitations	C-34
D	OS/32 AIDS GLOSSARY	D-1

TABLES

1-1	ADDRESS FIELDS	1-5
C-1	OS/32 AIDS COMMAND SUMMARY ENHANCEMENTS (CCM)	C-5

INDEX IND-1

PREFACE

This manual is a user's guide to OS/32 Automatic Interactive Debugging System (AIDS), a user-oriented assembly level debugging program intended for Perkin-Elmer 32-bit processors. Chapter 1 introduces basic functions, features, environment, address conventions and the directive syntax of OS/32 AIDS. Chapter 2 presents the various format option characters that are to be used within the directives. Chapter 3 covers the directives that are operational within OS/32 AIDS. Operating instructions, start procedures and error messages are discussed in Chapter 4. Appendixes A and B summarize OS/32 AIDS commands and present sample dump formats, respectively. Appendix C provides information regarding code compatible machine (CCM) support. Features of the OS/32 AIDS program that have been changed for CCM are detailed, along with discussions of the commands that have been altered. Appendix D is a glossary of general terms.

OS/32 AIDS User's Guide, 29-374 R04, has been superseded by this manual. Information contained within 29-374 R04 has been incorporated herein.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1
OS/32 AIDS

1.1 INTRODUCTION

OS/32 AIDS is a user-oriented assembly level debugging program for Perkin-Elmer 32-bit processors. AIDS debugs program user tasks (u-tasks) running in an operating system environment.

1.2 BASIC OPERATIONS

The OS/32 AIDS program can:

- display and modify memory locations, the floating point and general registers;
- print sections of memory to any physical device;
- provide program utilities such as snapshot printouts, cell/register protections, trace execution and breakpointing;
- provide single-step execution and display the current bias, location counter, task status or condition code (CC); and
- convert requested data or the current open location from one data format to another.

1.3 REQUIREMENTS

OS/32 AIDS requires the following system components:

- 19.25kb of memory
- 2,800 bytes of additional storage for each user
- an operating system of revision 02 or higher
- a print device
- a command device
- a binary output device

1.4 FEATURES

AIDS checks the operating system and sets the proper segmentation registers. It tests the system for single precision floating point and double precision floating point capabilities and sets flags to allow or disallow the floating point options available in AIDS. The task status word (TSW) is set so that AIDS processes supervisor call 14 (SVC14) calls, arithmetic, memory access and illegal instruction faults.

AIDS employs user-dedicated locations (UDLs) to accommodate the previously mentioned operations. These UDL locations are:

```
X'40' - X'4F'
X'60' - X'6F'
X'80' - X'9F'
X'C0' - X'CF'
X'F0' - X'FF'
```

When these faults occur, user registers 0 to 15 are saved in a writable area of the task.

1.5 MODES OF USE

AIDS can be used in either interactive or batch mode.

In interactive mode, AIDS reads a fixed format text line (directive) from the logical command unit and executes that directive. A prompt (>) is displayed whenever a command is to be entered on the operator's console. Commands and their options are accepted by AIDS in upper-case only. Error messages help the user to correct directive errors. The ability to display and modify the contents of memory locations, alter program flow, set breakpoints, protect data locations and execute single-steps are useful for interactive debugging.

When operating in batch mode, directive errors can result in a halt, which waits for operator intervention, or in an abort action, which cancels the task. When error conditions are encountered in batch mode, AIDS executes an end of job (EOJ) to return to the operating system.

Program utilities such as trace, snapshot dumps and protect capabilities make AIDS a useful debugging tool, either in interactive or batch mode.

1.6 DIRECTIVES

A directive is the text form of an AIDS command. Directives have a definite syntax beginning with a unique character opcode followed by a suitable option character. AIDS requires that these be in upper-case. If operand arguments are requested, they must be preceded by a space, separated by commas and ended with a carriage return (CR). The format of most directives is:

DO a_1,a_2,a_3

Parameters:

D	is the unique opcode character.
O	is the requested option.
a_1,a_2,a_3	are operands.

The following terms are used to explain the individual directives.

Address	is the location of a cell or register.
Option	specifies the data format of the directive (decimal, character, fullword hexadecimal, etc.).
Data	is used when the directive is executed.
Count	is a decimal field specifying a number of repetitions.
Delay	is a decimal field specifying the number of times a breakpoint or snapshot is encountered before being effected.

1.7 ADDRESS CONVENTIONS

An address field can have any of the following forms:

- Absolute — is the address field that is interpreted as the hexadecimal fixed address of a memory location within the user program space.
- Relative — is the address field that is added to the AIDS bias value to obtain the desired absolute address of a memory location.

- Local relative — is the address field that is added to the AIDS current location to obtain the address of the new desired location address.

- Registers — are the address fields that are interpreted as the hexadecimal number of a general or floating point register.

  - General purpose registers contain any hexadecimal digit from 0 to F preceded by a period (.).

  - Single precision floating point registers contain any even hexadecimal digit from 0 to E preceded by a colon (:).

  - Double precision floating point registers contain any even hexadecimal digit from 0 to E preceded by a semicolon (;).

When specifying memory locations, address fields can be hexadecimal numbers of up to six characters. An address field containing only a hexadecimal address is an absolute address. An address field followed by an asterisk (*) adds the current AIDS bias to the given hexadecimal address to give the desired absolute address. An address field preceded by an asterisk (*) and a plus sign (+) adds the address of the current open location to the given hexadecimal address to give the new open location address. When using AIDS in a multi-terminal monitor (MTM) environment, these addresses are relative to the task space.

Examples:

ADDRESS FIELD	MEANING
1A0	Open memory location at absolute address X'1A0'.
1A0*	Open memory location at absolute address X'1A0' plus the value of the current AIDS bias.
*+10	Increment the address of the current open location by X'10' and open that address.

If an address field is a register, the register number is preceded by the punctuation character period (.), colon (:) or semicolon (;). These characters indicate the general purpose, single precision floating point and double precision floating point registers, respectively.

Examples:

ADDRESS FIELD	MEANING
.A	GPR A
.2	GPR 2
:8	SPFPR 8
:E	SPFPR E
;C	DPFPR C
;4	DPFPR 4

The following examples are flagged as errors because the register addresses do not exist.

Examples:

ADDRESS FIELD	MEANING
.10	Flagged as NO-SUCH (10=16; no register 16)
:3	Flagged as NO-SUCH (odd number)
;9	Flagged as NO-SUCH (odd number)

Table 1-1 lists the address mode, mode indicator (key) and an example of each usage. A hexadecimal address value is represented by h.

TABLE 1-1 ADDRESS FIELDS

ADDRESS MODE	KEY	EXAMPLE	EXPLANATION
ABSOLUTE	h	32F0	32F0
RELATIVE	h*	1B0*	1B0 + BIAS value
LOCAL	*+h	*+10	Current address + 10
GPR	.h	.5	General purpose register 5
SPFPR	:h	:E	Single precision floating point register E
DPFPR	;h	;A	Double precision floating point register A

1.8 LOGICAL UNIT (LU) ASSIGNMENTS

AIDS uses three logical units for input/output (I/O). These units are assigned to the following within the program.

Command input unit	lu5	All directives are read from this unit.
List unit	lu3	Dump printed outputs and other printed outputs are listed to this unit.
Binary output unit	lu2	Loader format and binary dump output are written to this unit.

The command input device must be assigned prior to program execution. The output logical units must be assigned prior to their use.

The lu assignments for AIDS can be modified from the initial assignments by specifying options of the START command. See Section 4.2.3 for information on how to use the START command to change the default lu assignments. A second method, after the task has been started, is to use the LU command. See Section 3.13 to reassign logical units during execution.

CHAPTER 2
OS/32 AIDS DATA FORMAT MODES

2.1 INTRODUCTION

This section describes the data format modes that a directive can be requested to interpret. Data format modes are identified by a single option character. The following is a list of available data formats.

- Halfword hexadecimal
- Fullword hexadecimal
- Halfword decimal (integer)
- Fullword decimal (integer)
- Halfword character (ASCII)
- Single precision floating point
- Double precision floating point
- Assembly language

2.2 HALFWORD HEXADECIMAL FORMAT (X)

On input, AIDS accepts a maximum of four hexadecimal characters assumed to be right-justified. If more than four characters are entered, AIDS accepts the four right-most characters. Leading zeros can be omitted. Any character that is not a valid hexadecimal digit results in an error message. In all output, four hexadecimal digits are printed with leading zeros included.

2.3 FULLWORD HEXADECIMAL FORMAT (Y)

This fullword format is similar to the halfword hexadecimal format, except that up to eight hexadecimal digits are accepted on input, and eight hexadecimal digits are printed on output.

When using double precision (DP) registers, the Y mode displays the hexadecimal characters in doubleword format.

On input, AIDS accepts hexadecimal characters (0 to 9; A to F) and right-justifies them in the register specified.

NUMBER	PRINTED OUTPUT
A	0000 0000 0000 000A
ABCD	0000 0000 0000 ABCD
9ABCDEF	0000 0000 09AB CDEF
6789ABCDEF	0000 0067 89AB CDEF
3456789ABCDEF	0003 4567 89AB CDEF
F123456789ABCDEF	F123 4567 89AB CDEF

2.4 HALFWORD DECIMAL FORMAT (H)

On input, AIDS interprets the operand as a signed decimal integer. The plus sign (+) is assumed, but can be included. The minus sign (-) is required for negative numbers. An error message results if the magnitude is greater than 32,767. Leading zeros can be omitted. On output, the plus sign and leading zeros are omitted.

2.5 FULLWORD DECIMAL FORMAT (F)

This fullword format is the same as the halfword decimal format except that the magnitude can be as large as 2,147,483,647.

2.6 CHARACTER FORMAT (C)

On input, AIDS accepts a string of ASCII characters until a carriage return (CR) is encountered. A maximum of 60 characters is permitted. On output, each halfword is written to the output device as two ASCII characters with each nonprintable character converted to a period (.) before output.

2.7 SINGLE PRECISION FLOATING POINT FORMAT (E)

On input, AIDS accepts generalized floating point real number representation. Any single precision floating point number with a magnitude between .1 and 999999 is output without using exponential form. A minus sign is printed if the number is negative. Trailing zeros and decimal points are deleted. All other numbers are printed as follows.

Format:

(-).nnnnn(E±e)

Parameters:

n	is a decimal digit.
E	indicates single precision exponentiation.
e	is a 1- or 2-digit decimal number representing the exponent.
( )	indicate optional parts of the number.

Examples:

NUMBER	PRINTED OUTPUT
.00000002	.2E-7
-.0002	-.2E-3
200	200
-20,000,000,000	-.2E 11
00.0	0
-2.00	-2
12345678	.123456E8

Data will be normalized before output. If an invalid E or D number is encountered, the following message is displayed.

INVALID FLOATING POINT NUMBER

2.8 DOUBLE PRECISION FLOATING POINT FORMAT (D)

On input, AIDS accepts generalized double precision floating point real number representation. Any double precision floating point number with a magnitude between .1 and 99999999999999 is output without using exponential form. A minus sign is printed if the number is negative. Trailing zeros and decimal points are deleted. All other numbers are printed as follows:

Format:

(-).nnnnnnnnnnnnnn(D±e)

Parameters:

n	is a decimal digit.
D	indicates double precision exponentiation.
e	is a 1- or 2-digit decimal number representing the exponent.
( )	indicate optional parts of the number.

2.9 DISASSEMBLY FORMAT (A)

The disassembly format option is applicable to printed output only and disassembles machine language into assembler language. Extended branch mnemonics are used where they are implied. There may be some variations of the extended mnemonics entered by the programmer and those disassembled by AIDS because some extended branches have identical opcodes for different mnemonics, i.e. BES, BZS.

Output Format:

(Address) (Contents) (Mnemonic) (Operands)

Parameters:

Address	is a 6-digit hexadecimal address either absolute or relocatable.
Contents	are one, two or three halfwords of machine code displayed in hexadecimal digits.
Mnemonic	is the appropriate mnemonic for the machine instruction.
Operands	are the addresses or literal operands for the displayed instruction with appropriate index registers. The display is always hexadecimal information.

Example:

ADDRESS	CONTENTS	MNEMONIC	OPERANDS
00175CR	48DF 0002	LH	D,2(F)

The disassembler will attempt to disassemble any and all input regardless of its meaning.

The starting address must be the first halfword of the instruction. The disassembler may not describe all data as data, but instead as an instruction. AIDS will try to assemble any define constants or storage.

Example:

```
            "TRUE CODE"
START       LH 0,DATA
            SVC 3,0
DATA        DC X'2401'

            AIDS DISASSEMBLER
            LH 0, ADDRESS


            SVC 3,0
            LIS 0,1
```

CHAPTER 3
OS/32 AIDS COMMANDS

3.1 INTRODUCTION

AIDS commands are presented in alphabetical order within this chapter. A list of all AIDS commands follows:

ADD	GO	LOG	OPEN PREVIOUS
BATCH	INSERT	LOGICAL UNIT	PAUSE
BIAS	Breakpoint	NEXT	REPLACE
CONVERT	Protect	NO BATCH	SUBTRACT
DUMP	Snapshot Dump	NO LOG	TYPE
END	Trace	OPEN	WHERE
EXECUTE	JUMP	OPEN NEXT	ZAP

The functional description of all AIDS directives and their syntax with legal options and operands are identified in the following sections. Usage notes and cautions are given in each section where required. All directives are input followed by a carriage return (CR) when in the interactive mode. All directives and their options must be in upper-case to be acceptable to AIDS. Most directives are recognized by a single input character. Options are represented by the second character position without any intervening characters or spaces.

NOTE

Users who can use the code compatible machine (CCM) instruction set should see Appendix C for information relating to CCM support features. Specific AIDS commands have been altered to be consistent with CCM support. The CCM instruction set, along with the EBCDIC character set, is now allowed and is reflected in the revised option parameters (see Section C.3.8).

| ADD |

3.2 ADD COMMAND

This command increases the contents of the current open location or register by a given value if that value is given in the operand field ($data_1$). The value must be in the valid format identified by the option field.

Format:

A[option] $data_1$ [,$data_2$]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- D — specifies that the contents of the current open location or register is in the double precision floating point format.
- E — specifies that the contents of the current open location or register is in the single precision floating point format.
- F — specifies that the contents of the current open location or register is in the fullword decimal format.
- H — specifies that the contents of the current open location or register is in the halfword decimal format.
- X — specifies that the contents of the current open location or register is in the halfword hexadecimal format.
- Y — specifies that the contents of the current open location or register is in the fullword hexadecimal format.

$data_1$ — specifies the value to be added to the contents of the current open location or register if $data_2$ is omitted.

$data_2$ specifies the value to be added to $data_1$. The results of the two added values are displayed.

Functional Details:

A convenient feature is provided by the second operand option. When the user specifies $data_2$, AIDS adds the two data values in the specified mode and displays the results. AIDS does not update memory or change the current open mode as a result of this calculation. Halfword modes are not valid with $data_2$ entry.

Examples:

Increase the contents of single precision floating point register 6 by 5.31.

```
>OE :6
 FP (6)      10
>AE  5.31
>OE :6
 FP (6)      15.31
```

Add two hexadecimal numbers.

```
>AY  4803051E,100F
     4803152D
```

NOTE

The address of the current open location is not incremented to the next logical location or register in either the ADD or SUBTRACT directive.

3.3 BATCH COMMAND

If a command error occurs, the BATCH command terminates the job and returns control to the operating system.

Format:

```
B
```

Functional Details:

The BATCH command is used when directives are in a command file and operator intervention is not practical.

3.4 BIAS COMMAND

This command sets or displays the address value contained in the memory location BIASVAL, which holds the bias value within AIDS.

Format:

```
BI [address]
```

Parameters:

address	is a 1- to 6-digit hexadecimal number specifying the value to which the bias is set. If this parameter is omitted, the current value of the bias is displayed.

Functional Details:

When the bias is set to a value, any use of the relative address option causes the bias value to be added to the input, when the address is input, and subtracted in the case of relative addresses when displayed by AIDS.

The bias will only be added to those addresses followed by an asterisk (*). This facilitates debugging a main segment using the absolute address and debugging a subroutine segment that was separately assembled.

Each segment assembly listing starts at relative 0; therefore, setting the bias value to the link loading address for the subroutine allows references to locations with that segment without doing the hexadecimal calculations.

Examples:

Set the bias to X'6A0'.

```
>BI 6A0
```

Display the current value of the bias.

```
>BI
0006A0
```

3.5 CONVERT COMMAND

This command converts fullword data from one data format to another.

Format:

C [option] data

Parameters:

option — specifies the type of conversion with the choice of one of the following parameters:

- CY — specifies that data is to be converted from character to hexadecimal format.
- YC — specifies that data is to be converted from hexadecimal to character format.
- FY — specifies that data is to be converted from fullword decimal to hexadecimal format.
- YF — specifies that data is to be converted from hexadecimal to fullword decimal format.
- EY — specifies that data is to be converted from single precision floating point to hexadecimal format.
- YE — specifies that data is to be converted from hexadecimal to single precision floating point format.
- DY — specifies that data is to be converted from double precision floating point to hexadecimal format.
- YD — specifies that data is to be converted from hexadecimal to double precision floating point format.

data — specifies the value to be converted.

Functional Details:

For the user to convert data from one format to another, the only allowable conversion operations are the following:

Character to hexadecimal	C→Y
Hexadecimal to character	Y→C
Decimal to hexadecimal	F→Y
Hexadecimal to decimal	Y→F
Single precision floating point to hexadecimal	E→Y
Hexadecimal to single precision floating point	Y→E
Double precision floating point to hexadecimal	D→Y
Hexadecimal to double precision floating point	Y→D

The execution of this command does not alter any memory or mode options within AIDS. Halfword modes are not applicable.

Examples:

Hexadecimal to character:

```
>CYC 44
    D
```

Single precision floating point to hexadecimal:

```
>CEY 10.0
 41A00000
```

Hexadecimal to single precision floating point:

```
>CYE 41A00000
     10
```

Decimal to hexadecimal:

```
>CFY 2858
     00000B2A
```

Hexadecimal to decimal:

```
>CYF B2A
     2858
```

Double precision floating point to hexadecimal:

```
>CDY -.67489226182806D47
      E7BD2525 24C52E6C
```

Hexadecimal to double precision floating point:

```
>CYD E7BD2525 24C52E6C
     -.67489226182806D47
```

3.6 DUMP COMMAND

This command outputs the contents of a section of memory defined by the starting and ending address parameters.

Format:

D [option] address$_1$,address$_2$

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output is to be displayed in assembly language format.
- B — specifies that the output is to be written to logical unit 2 (lu2) in memory image binary format.
- C — specifies that the printed output is to be displayed in character format.
- D — specifies that the printed output is to be displayed in double precision floating point format.
- E — specifies that the printed output is to be displayed in single precision floating point format.
- F — specifies that the printed output is to be displayed in fullword decimal format.
- H — specifies that the printed output is to be displayed in halfword decimal format.
- L — specifies that the output is to be written to lu2 in loader (binary object) format (executable form).

U specifies that the printed output is to be displayed as a listing of all breakpoint locations within the dump.

X specifies that the printed output is to be displayed in halfword hexadecimal format.

Y specifies that the printed output is to be displayed in fullword hexadecimal format.

$address_1$ is a 1- to 6-digit hexadecimal number specifying the starting address of the contents to be dumped.

$address_2$ is a 1- to 6-digit hexadecimal number specifying the ending address of the contents to be dumped.

Functional Details:

The DUMP command outputs the contents of a section of memory defined by two address operands. The first address is the start address and the second address is the ending address of the requested section of memory to be output. The format of the dump output is specified by the option character in the directive.

The existence of active sentinels within user programs causes erroneous dumps. When this occurs, remove all sentinels through the ZAP command before dumping.

The DUMP command displays the address locations of all set program sentinels. If a bias is set in AIDS, and the sentinel address location is to add the bias, the address locations are printed relative to the bias and are indicated by the character R after the address location.

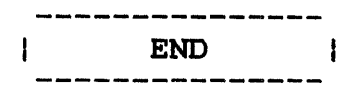
The output of all the dump parameters is written to lu3, except the loader format (L) and the memory image binary (B), which are written to lu2.

When the output is in memory image binary format (B), AIDS dumps only one record per command with an indexed file. However, when using a contiguous file, an error message will appear indicating that space is unavailable for this operation.

See Appendix B for examples of dump formats written to lu3.

The following is a list of the DUMP command data formats:

TYPE	OPTION
Disassembly	A
Memory image binary	B
Character	C
Double precision floating point	D
Single precision floating point	E
Fullword decimal	F
Halfword decimal	H
Loader	L
Listing of sentinel addresses	U
Halfword hexadecimal	X
Fullword hexadecimal	Y



3.7 END COMMAND

This command terminates AIDS processing and returns control to the operating system with end of task code 0.

Format:

```
EN
```

3.8 EXECUTE COMMAND

This command executes a given number of instructions.

Format:

X[T] [count] [,address]

Parameters:

T	specifies that each instruction and its address will be displayed to the list device assigned to the output unit upon execution.
count	specifies the number of instructions to be executed. If this parameter is omitted, the default is one.
address	is a 1- to 6-digit hexadecimal number specifying the starting address of the instructions to be executed. If this parameter is omitted, the default is the address of the last instruction executed.

Functional Details:

A breakpoint causes a return to the AIDS command level regardless of the number of instructions to be executed. The instruction address location is printed relative to the bias if the address is greater than the bias.

Instructions in a shared task segment or in a write-protected partial image segment cannot be executed by this command.

Examples:

Execute and print four instructions starting at address location X'1A0' with no bias set.

```
>XT 4,1A0

 0001A0: 7820 FF74      LD      2,118
 0001A4: 3842           LDR     4,2
 0001A6: 7F60 FF8E      LMD     6,138
 0001AA: 7E60 FFBA      STMD    6,168
```

Execute and print two instructions starting at address location X'1A0' with the bias set to X'100'.

```
>BI 100
>XT 2,1A0

 0000A0R 7830 FF74      LD      2,118
 0000A4R 3842           LDR     4,2
```

Execute the next logical instruction without printing the instructions.

```
>X
```

Execute and print the next instruction with the bias set to X'100'.

```
>XT

 0000AAR 7E60 FFBA      STMD    6,168
```

NOTE

The AIDS location counter always points to the next logical executable instruction.

3.9 GO COMMAND

This command starts or returns processing to a user program with the machine state and condition code restored to its initial or interrupted value.

Format:

GO [address] [,user start options]

Parameters:

address	is a 1- to 6-digit hexadecimal number specifying the location at which execution of the GO command starts. If this parameter is omitted, execution continues with the next logical instruction following the last specified sentinel encountered (P, S, T, X). The default value of the address field is initially X'100'.
user start options	specifies any options the user program needs to execute correctly, such as logical units, files or devices. A maximum of eighty characters is allowed.

Functional Details:

If only an address field is specified, AIDS places a carriage return at the current UTOP.

If user start options are specified, all characters from the one immediately following the comma to the carriage return (80 characters maximum) move to current UTOP.

If the GO command is a continuation following a breakpoint, the instruction at the breakpoint location is executed and normal execution proceeds. The breakpoint is not removed.

Examples:

Execute program at address X'3000'.

```
>GO 3000
```

Execute program at current open location plus 10.

```
>GO *+10
```

Continue executing from last breakpoint.

```
>GO
```

Execute program at address X'100' after placing options at UTOP for user.

```
>GO 100,IN=D0,OUT=D1,LIST=PR:
```

Restart execution at address 1D0 plus the bias value after modifying specific memory locations and/or register locations.

```
>GO 1D0*
```

3.10 INSERT COMMAND

This command requires one of the following AIDS operations:

- Breakpoint (X)
- Protect (P)
- Trace (T)
- Snapshot Dump (S)

Format:

$$I\begin{cases} X \\ P \\ T \\ S \end{cases}\begin{cases} [*]\ \text{xaddress}\ [*]\ [,\text{delay}] \\ [H]\ [*]\ \text{paddress}\ [*] \\ [B]\ [*]\ \text{taddress}_1\ [*],\text{taddress}_2\ [*] \\ [\text{option}]\ [*]\ \text{address}_1\ [*],\text{address}_2\ [*]\ [,\text{delay}] \end{cases}$$

Parameters:

X	specifies a breakpoint. A breakpoint halts program execution at a specified address location. A maximum of 32 breakpoints can be maintained.
*	specifies that the address field is relative to the bias set in AIDS and identified with the character 'R' following the address location. If this parameter is omitted, the displayed address field defaults to absolute.
xaddress	is a 1- to 6-digit hexadecimal number specifying the location at which program execution is halted.

delay — specifies the number of times a breakpoint is encountered before program execution is halted. The delay request must not exceed 32,767. If the delay field is omitted, a delay of zero is assumed and execution halts each time the breakpoint sentinel is encountered. When a delayed breakpoint is activated, the delay count is reset and does not reactivate until the delay count is reached again.

P — specifies protect. Use of this parameter protects single precision floating point, double precision floating point, general purpose registers or memory locations. A maximum of 32 protect data formats can be maintained.

H — specifies that a halfword location is to be protected. If this parameter is omitted, fullword protection is assumed.

paddress — is a 1- to 6-digit hexadecimal number specifying the register or memory location to be protected.

T — specifies trace. Trace monitors and displays each time specified instructions in a program are executed. A maximum of 32 trace regions can be defined.

B — specifies branch trace. This parameter displays the value of the current location when a branch instruction is encountered and executed.

$taddress_1$ — is a 1- to 6-digit hexadecimal number specifying the starting location of the trace region.

$taddress_2$ — specifies the final address of the trace region.

S — specifies snapshot dump. Snapshot dump displays specified memory locations and registers at a selected location during program execution. A maximum of 32 snapshot sentinels can be inserted.

option — specifies the type of format in effect with the choice of one of the following parameters:

A — specifies that the contents to be dumped will be displayed in assembly language format.

B specifies that the contents to be dumped will be displayed in memory image binary format.

C specifies that the contents to be dumped will be displayed in character format.

D specifies that the contents to be dumped will be displayed in double precision floating point format.

E specifies that the contents to be dumped will be displayed in single precision floating point format.

F specifies that the contents to be dumped will be displayed in fullword decimal format.

H specifies that the contents to be dumped will be displayed in halfword decimal format.

L specifies that the contents to be dumped will be displayed in loader (binary object) format.

X specifies that the contents to be dumped will be displayed in halfword hexadecimal format.

Y specifies that the contents to be dumped will be displayed in fullword hexadecimal format.

$address_1$ is a 1- to 6-digit hexadecimal number specifying the location of the start of the snapshot dump.

$address_2$ is a 1- to 6-digit hexadecimal number specifying the ending address of the snapshot dump.

Functional Details:

A maximum of 32 breakpoint, protect, trace or snapshot sentinels can be maintained.

Inserting a breakpoint means AIDS inserts the instruction X'E1E0' (SVC 14,0) at the requested address in the user code. The replaced instruction is saved in the breakpoint table and restored when the breakpoint is cleared. Executing a breakpoint does not clear the breakpoint. When a breakpoint is encountered, the following message is displayed:

BP (address)

Breakpoints must be inserted on the first halfword of an instruction. An error message results if a breakpoint is inserted on an existing breakpoint.

Breakpoints cannot be inserted into shared task segments or write-protected partial image segments. Inserting a breakpoint into a write-enabled partial image segment has an undefined effect upon other tasks using that segment.

Protection of a register or memory location does not modify the user program. The value of the register or memory location is saved by AIDS at the time the protect option is specified. When the AIDS interpreter executes each user instruction, each protected item value is checked with the stored value in the protect table. If the value of the protected item changed, then the new value is saved and the following message is output:

(address$_1$) CH (address$_2$) FR (data$_1$) to (data$_2$)

Address$_1$ is the address location of the instruction that caused the contents of address$_2$ (the protected item) to change from data$_1$ to data$_2$. All data is displayed in hexadecimal, but the address can be relative or absolute. To protect double precision locations other than registers, both words must be protected individually.

The TRACE command lists the present value of the location counter and the disassembly of the instruction, while only the location of the branch instruction and its destination address are listed under branch trace.

An error message results if overlapping trace areas are specified. There are no sentinels inserted in the user program from the trace feature.

Delays are available on each snapshot and more than one type of dump is available simultaneously on a single snapshot. Snapshot dumps are identical with normal dump outputs.

Setting a snapshot involves inserting an SVC 14,0 at the current open location or register. For this reason, AIDS does not permit the user to insert a breakpoint over a snapshot or vice versa.

Examples:

Set a breakpoint at X'3519E' with no delay.

```
>IX 3519E
```

Set a delayed breakpoint at X'51E' with a delay of 10.

```
>IX 51E,10
```

Set a delayed breakpoint at the current location +20 and display the absolute address.

```
>IX *+10,150
```

Set a breakpoint at relative address 50 and display in relative format.

```
>IX* 50*
```

Set a breakpoint at relative address 7E and display in relative format with a delay of 20.

```
>IX* 7E*,20
```

Protect single precision floating point register C.

```
>IP :C
 00365E: CH FP(C)FR 413896E1 to 42000000
```

Protect location 100 relocatable and display in relative format.

```
>IP* 100*
 001F32R CH 000100R FR 0000385E to 1000385E
```

Protect double precision floating point register 8.

```
>IP ;8
 0006A0:  CH DF (8) FR E7BD2525 24C52E6C to 42000000 00000000
```

Set a trace region between X'1A0' and X'1E0' and start execution at 1A0.

```
>IT 1A0,1E0
>GO 1A0

 0001A0:  7820 FF74                    LD      2,118
 0001A4:  3742                         LDR     4,2
 0001A6:  7E60 FF8E                    LMD     6,138
 0001AA:  7E60 FFBA                    STMD    6,168
 0001Ae:  7040 FFDE                    STD     4,190
 0001B2:  3942                         CDR     4,2
 0001B4:  4330 4000 01C8               BE      1C8
 0001C8:  7960 FF9C                    CD      6,168
 0001CC:  4330 4000 01E0               BE      1E0
 0001E0:  2418                         LIS     1,8
```

Set a trace region between X'1B2' and X'200' and trace relative locations. Assume a bias set to X'100'.

```
>IT* 1B2,200
>GO  1A0

 0000B2R 3942                          CDR     4,2
 0000B4R 4330 4000 01C8                BE      1C8
 0000C8R 7950 FF9C                     CD      6,168
 0000CCR 4330 4000 01E0                BE      1E0
 0000E0R 2418                          LIS     1,8
 0000E2R 7980 FF5A                     CD      8,140
 0000E6R 4330 4000 01FA                BE      1FA
 0000FAR 79A1 4100 0168                CD      A,168(1,1)
 000100R 4330 4000 0214                BE      214
```

Dump memory from X'100' to X'200' in single precision floating point with no delay, when location 400 is reached. Open location 400 and insert a snapshot dump to display address 100 through address 200 each time the instruction at address 400 is executed.

```
>OH 400
>ISE 100,200
```

Dump memory from current open location to current open location +100 after delaying 50 times. Dump in relocatable assembly language. Display addresses in relative format. The bias is subtracted.

```
>ISA* *,*+100,50
```

Dump memory in binary format from X'40000' to X'42000' each time the current open location is encountered. Output is written to lu2 for loader and binary format dumps.

```
>ISB 40000,42000
```

Dump all the general purpose registers in fullword hexadecimal when the instruction at location 1000 is encountered for the 25th time.

```
>OH 1000
>ISY .0,.F,25
```

Dump all the general purpose registers in fullword hexadecimal every 10 times the instruction at 1000 is encountered. Also dump single precision floating point registers 2 through 8 every 20 times the same instruction is encountered.

```
>OH 1000
>ISY .0,.F,10
>ISE :2,:8,20
```

3.11 JUMP COMMAND

This command displays and opens a new location pointed to by the contents of the current open location.

Format:

J [option]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the new current open location will be displayed in assembly language format.
- C — specifies that the new current open location will be displayed in character format.
- D — specifies that the new current open location will be displayed in double precision floating point format.
- E — specifies that the new current open location will be displayed in single precision floating point format.
- F — specifies that the new current open location will be displayed in fullword decimal format.
- H — specifies that the new current open location will be displayed in halfword decimal format.
- X — specifies that the new current open location will be displayed in halfword hexadecimal format.
- Y — specifies that the new current open location will be displayed in fullword hexadecimal format.

Functional Details:

If no data format parameters are specified, the current open mode is used as the default. The new open location becomes the new current open location, and the contents of the location pointed to by the current open location are interpreted as a fullword address.

Example:

Display the contents of the location pointed to by the contents of the current open location.

```
>OX  500
 000500:   3140
>JX
 003140:   1AE
```

3.12 LOG COMMAND

This command copies the input command stream and error messages to the list device when used in a batch environment.

Format:

```
L
```

3.13 LOGICAL UNIT (LU) COMMAND

This command reassigns the lu assignments for AIDS if there is a conflict with the logical units used by the program to be debugged.

Format:

LU U_1,U_2,U_3

Parameters:

U_1	is a decimal number from 1 through 254 specifying the command input device. The default assignment for the command input device is lu5.
U_2	is a decimal number from 1 through 254 specifying the list device. The default assignment for the list device is lu3.
U_3	is a decimal number from 1 through 254 specifying the binary output device. The default assignment for the binary output device is lu2.

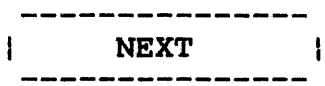
Functional Details:

The LU command is used with three required (nondefault) parameters. Use the operating system ASSIGN command to establish the correct physical device assignments prior to using the LU command, or change the logical units when AIDS is started.

Example:

Reassign the list output unit to lu4, and the binary output unit to lu6.

```
>LU 5,4,6
```



3.14 NEXT COMMAND

This command causes the next logical location to be opened, but not displayed. This location becomes the new current location.

Format:

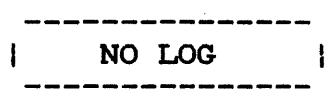
N

3.15 NO BATCH COMMAND

This command cancels the batch mode.

Format:

NB



3.16 NO LOG COMMAND

This command cancels the LOG command. AIDS is initialized to the NO LOG mode.

Format:

NL

3.17 OPEN COMMAND

This command displays the contents of the location or register in any of the data formats described in Chapter 2. When the location is opened, the address and mode become the current open location and the current open mode.

Format:

O[option] address[,count]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output will be displayed in assembly language format.
- C — specifies that the printed output will be displayed in character format.
- D — specifies that the printed output will be displayed in double precision floating point format.
- E — specifies that the printed output will be displayed in single precision floating point format.
- F — specifies that the printed output will be displayed in fullword decimal format.
- H — specifies that the printed output will be displayed in halfword decimal format.
- X — specifies that the printed output will be displayed in halfword hexadecimal format.
- Y — specifies that the printed output will be displayed in fullword hexadecimal format.

address — is a 1- to 6-digit hexadecimal number specifying a relocatable or absolute address.

count — is a decimal number specifying the number of locations or registers to be opened and displayed beyond the indicated address. If this parameter is omitted, a default of one is assumed.

Examples:

Open location 1004 and display in disassembly format for three consecutive locations.

```
>OA 1004,3

 001004:   0822       LR    2,2
 001006:   43308062   BZ    106C
 00100A:   C8300045   LH1   3,45
```

Open location 1006 and display in character format.

```
>OC 1006

 001006:   C0
```

Open double precision register A and display in fullword hexadecimal format.

```
>OY ;A

 DF (A)    43AC1000 76390000
```

Open single precision floating point register C and display as floating point.

```
>OE :C

 FP (C)    10
```

Open general purpose registers 5 through 7 and display as fullword decimal.

```
>OF .5,3

 GP(5)     13
 GP(6)     901437
 GP(7)     -213
```

Open location 21F0 + 10 in halfword hexadecimal with the current open location at address 0021F0.

```
>0 *+10

 002200:  FE91
```

Open relative location 1000 in character mode with the bias value set to 2000.

```
>OC 1000*

 003000: EB
```

NOTE

An asterisk (*) specifies that the address field is relative. The character R is printed after the address location.

```
OC* 1000*

001000R  EB
```

3.18 OPEN NEXT COMMAND

This command displays the next logical location or register in the format specified by the format option character.

Format:

$$\left\{\begin{matrix}\text{spacebar} \\ \text{cr}\end{matrix}\right\}[\text{option}]$$

Parameters:

spacebar — specifies that the spacebar can be pressed to display the next location or register in the format that was previously in effect. If followed by an option character, the display will be altered to the format specified.

CR — specifies that a carriage return entered alone will display the next location or register in the format that was previously in effect.

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output of the next location or register will be displayed in assembly language format.
- C — specifies that the next location or register will be displayed in character format.
- D — specifies that the next location or register will be displayed in double precision floating point format.
- E — specifies that the next location register will be displayed in single precision floating point format.
- F — specifies that the next location or register will be displayed in fullword decimal format.

H specifies that the next location or register will be displayed in halfword decimal format.

X specifies that the next location or register will be displayed in halfword hexadecimal format.

Y specifies that the next location or register will be displayed in fullword hexadecimal format.

Functional Details:

The format of this command is a carriage return (CR) or a space followed by a data format parameter. If a data format parameter is not specified, the last one specified is used to display the location.

Example:

```
>OH 1000
 001000:   4120
>(Space)
 001002:    8830
>OF .5
 GP (5)     10
>Carriage return
 GP (6)     20141
>Carriage return
 GP (7)     0
```

3.19 OPEN PREVIOUS COMMAND

This command displays the location or register immediately preceding the current open location or register in the format specified by the format option parameter.

Format:

-[option]

Parameters:

- specifies that the first portion of the format is a negative sign (-).

option specifies the type of format in effect with the choice of one of the following parameters.

A specifies that the preceding location or register will be displayed in assembly language format.

C specifies that the preceding location or register will be displayed in character format.

D specifies that the preceding location or register will be displayed in double precision floating point format.

E specifies that the preceding location or register will be displayed in single precision floating point format.

F specifies that the preceding location or register will be displayed in fullword decimal format.

H specifies that the preceding location or register will be displayed in halfword decimal format.

X specifies that the preceding location or register will be displayed in halfword hexadecimal format.

Y specifies that the preceding location or register will be displayed in fullword hexadecimal format.

Functional Details:

If no data format parameter is specified, the last format specified is used.

Example:

```
>OH 1000
 001000:   31F0
>-
 000FFE:   389A
>OF .7
 GP (7)    0
>-
 GP (6)    20141
>-
 GP (5)    10
```

| PAUSE |

3.20 PAUSE COMMAND

This command allows an exit to the operating system. The operating system CONTINUE command can be used to return to AIDS.

Format:

```
P
```

Functional Details:

While AIDS is operating, the user can pause its operation and perform other functions, or obtain other information, by a return to the operating system. This is done by using the PAUSE command.

The occurrence of some faults will cause AIDS to pause. The message "task paused" will appear on the terminal below a brief description of the fault.

In order to have AIDS resume operation, use the CONTINUE command along with the restart address, if necessary. The AIDS restart address equals starting address + 4.

3.21 REPLACE COMMAND

This command replaces the contents of the current open location or register with new data specified in the operand field of the directive.

Format:

R [option] data [,count]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- C — specifies that the new contents consist of a 1- to 60-character ASCII string.
- D — specifies that the new contents are set to double precision floating point format.
- E — specifies that the new contents are set to single precision floating point format.
- F — specifies that the new contents will be displayed as a 1- to 10-digit fullword decimal format.
- H — specifies that the new contents will be displayed as a 1- to 5-digit halfword decimal number.
- X — specifies that the new contents consist of a 1- to 4-digit hexadecimal number.
- Y — specifies that the new contents consist of a 1- to 8-digit hexadecimal number.

data — specifies the data to be used as the new contents.

count — specifies the number of locations or registers being replaced with new data. A contiguous block of memory can be initialized to a specific value through the use of the count option. The count value defaults to 1.

Functional Details:

Execution of the REPLACE command automatically increments the current open location to the next logical location (i.e., the amount of memory operated on by the REPLACE command). This enables the user to conveniently replace memory without opening the next location each time.

If a parameter is not specified, the option currently in effect is used to interpret the operand field.

Data in shared task segments or write-protected memory cannot be modified by this command.

Examples:

Set all double precision floating point registers to 0.

```
>OD ;0          Sets the current open location to double
                precision floating point register (0).
 DF (0)         -40.156254907208
>R 0,8          Replaces the contents of eight consecutive
                registers with zero starting with the current
                open register, which is the double precision
                floating point register 0.
```

Set single precision floating point register A to 385.671.

```
>OE :A          Opens single precision floating point register
                (A).
 FP (A)         10
>R  385.671     Replaces the contents of the single precision
                floating point register A with the value.
```

Set all fullwords in memory from location X'1000' to location X'2000' to a value of -1.

```
>OF 1000        Opens memory location X'1000'.
 001000:        13501
>R -1,2048      Replaces 2048 fullwords with the fullword
                decimal value -1.
```

Set the value of the current open location to halfword decimal 10353.

```
>RH 10353
```

Store the characters ABCD in the current open fullword or current open register.

```
>RC ABCD
```

NOTE

When replacing in the character mode, be careful not to overwrite memory beyond the desired location. AIDS accepts a string of up to 60 characters or up to a carriage return. It is, therefore, possible to enter a string larger than the anticipated space.

3.22 SUBTRACT COMMAND

This command decreases the contents of the current open location or register by a given value. This operation is similar to the ADD directive except subtraction, rather than addition, occurs.

Format:

S[option] $data_1$ [,$data_2$]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- D — specifies that the contents of the current open location or register are in double precision floating point format.
- E — specifies that the contents of the current open location or register are in single precision floating point format.
- F — specifies that the contents of the current open location or register are in fullword decimal format.
- H — specifies that the contents of the current open location or register are in halfword decimal format.
- X — specifies that the contents of the current open location or register are in halfword hexadecimal format.
- Y — specifies that the contents of the current open location or register are in fullword hexadecimal format.

$data_1$ — specifies the value to be subtracted from the contents of the current open location or register.

$data_2$ — specifies the value from which data is to be subtracted.

Functional Details:

The address of the current open location is not increased to the next location or register in either the ADD or SUBTRACT directive.

Examples:

Decrease the contents of general purpose register 5 by 10.

```
>OF .5
 GP (5)     23
>SF 10
>OF .5
 GP (5)     13
```

Subtract two hexadecimal numbers. Subtract 29FB from 135EDA.

```
>SY 29FB, 135EDA
 001334DF
```

| TYPE |

3.23 TYPE COMMAND

This command displays and allows the updating of the user's task status word (TSW) or the condition code.

Format:

$$T\left\{\begin{matrix}S\\C\end{matrix}\right\}$$

Parameters:

S	displays the status portion of the user's TSW. The information displayed consists of bits 0 through 31 of the 64-bit TSW only.
C	displays the current program condition code. The information displayed consists of bits 28 through 31 of the 64-bit TSW.

Functional Details:

A new status portion of the TSW can be entered by the REPLACE command. Since this affects internal AIDS tables, the use of commands that increment the current location is not immediately allowed after the REPLACE command. Entry is in fullword hexadecimal format and does not have to be user-specified.

A new condition code can be entered by the REPLACE command. Entry is in halfword hexadecimal format and does not have to be user-specified.

Examples:

Display the current TSW and replace it with the new TSW 06007010.

```
>TS
   00007210
>R 6007010
>TS
   06007010
```

Display the current condition code and set it to X'2'.

```
>TC
   3
>R 2
```

3.24 WHERE COMMAND

This command prints the current value of the location counter in AIDS.

Format:

```
W
```

Functional Details:

If a bias was set and the address of the location counter is greater than the bias, the printed value is relative to the set bias.

Examples:

Print the current address of the location counter, when the AIDS bias is zero.

```
>W
  000476:
```

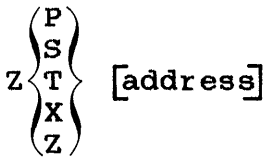
Print the current address of the location counter with the AIDS bias set.

```
>BI 100
>W
  000376R
```

3.25 ZAP COMMAND

This command removes the sentinel that was previously set.

Format:



Parameters:

P	specifies that the protect sentinel is to be removed.
S	specifies that the snapshot sentinel is to be removed.
T	specifies that the trace sentinel is to be removed.
X	specifies that the breakpoint sentinel is to be removed.
Z	specifies that all sentinels are to be removed.
address	is a 1- to 6-digit hexadecimal number specifying the specific sentinel to be removed. If this parameter is omitted, all sentinels of the specified type (protect, snapshot, trace, breakpoint or all) are removed.

Examples:

Unprotect double precision floating point register E.

```
>ZP ;E
```

Unprotect memory location D00.

```
>ZP D00
```

Remove the trace region from X'3000' to '3500'.

```
>ZT 3000
```

Remove all breakpoint sentinels.

```
>ZX
```

Remove all sentinels of all classes.

```
>ZZ
```

CHAPTER 4
OS/32 AIDS OPERATING INSTRUCTIONS

4.1 OS/32 AIDS OPERATION

User-dedicated locations (UDLs) are used by AIDS to accommodate supervisor call 14 (SVC14), arithmetic faults, memory access faults and illegal instruction faults. These UDLs are:

```
X'40' - X'4F'
X'60' - X'6F'
X'80' - X'9F'
X'C0' - X'CF'
X'F0' - X'FF'
```

Be aware that these locations are subject to change. When these faults occur, registers are saved at user registers 0 to 15 in an impure area.

AIDS automatically tests for single precision floating point and double precision floating point capabilities and sets flags to allow or disallow the floating point features. The task status word (TSW) is set for SVC14, arithmetic fault, memory access fault and illegal instruction fault. Option AFC must be specified when the program is established using OS/32 Link, if the arithmetic fault is to operate properly and return control to AIDS.

4.2 START PROCEDURES

AIDS can be made available to the user in two ways:

- By using OS/32 Link to establish AIDS as part of a user task (u-task) or executive task (e-task).
- By using OS/32 Link to establish AIDS as a shared partial image segment and attach it to a u-task. This method is limited to u-tasks only.

4.2.1 Building AIDS into a User Task (U-Task)

Link can establish AIDS as part of a u-task by including it at establishment time with the INCLUDE command. The following sequence of commands is to be used:

```
*LO LINK                    Loads Link.
*ST                         Starts Link.

-PERKIN-ELMER OS/32 LINKAGE EDITOR 03-242 R01-00

ESTABLISH TASK              Establishes the task.
OPTION WORK=7000            Requests additional storage.
INCLUDE TEST[.OBJ]          Includes the program.
INCLUDE OSAIDS.[.OBJ]       Includes AIDS.
OPTION NSEGMENTED           Entire task must be impure.
MAP fd:,ADDRESS             Generates summary and address map.
BUILD TEST                  Builds the task.
END                         End of job.
```

AIDS will debug either u-tasks or e-tasks when included as part of the task. To start AIDS as part of the u-task, issue the START command, giving the origin address of AIDS within the task space. Additional storage must be requested. The command OPTION WORK=XB00 allocates the proper additional space required.

If OSAIDS.OBJ is included last in the above sequence, its starting address will be the main starting address of the task.

See the OS/32 Link Reference Manual for detailed information about the use of each Link command.

4.2.2 Building AIDS as a Shared Partial Segment

AIDS can be established and used as a partial image segment by using OS/32 Link commands. A partial image is a collection of task segments that can be used by one or more separate tasks. A partial image has no UDL. As a partial image segment, AIDS executes an SVC 2,2 to get storage of X'600' bytes for a work area. A task to be debugged with AIDS should make allowance for the required area with the Link OPTION WORK= command. As a shared partial segment, AIDS debugs only u-tasks.

The following sequence of OS/32 Link commands are to be used to establish the shared partial image for multiple users:

```
*LO LINK                    Loads Link.
*ST                         Starts Link.

-PERKIN-ELMER OS/32 LINKAGE EDITOR 03-242 R01-00

>ESTABLISH IMAGE,ADDRESS=XXXX,ACCESS=RE
>INCLUDE OSAIDS[.OBJ]      Includes OS/32 AIDS within image.
>MAP fd:                   Sends establishment summary to
                           the specified device.
>BUILD OSAIDS              Starts building image.
END                         End of job

USER       -END OF TASK CODE=  2  CPUTIME=1.050/0.403
```

When building the shared partial image segment, the ESTABLISH command must specify XXXX as a real address.

To access the shared partial image segment that has been built, use the following OS/32 Link commands when linking the task to be debugged:

```
ESTABLISH TASK
OPTION WORK=7000
INCLUDE TEST
RESOLVE OSAIDS
MAP fd:,ADDR
BUILD TEST
END
```

To start AIDS when established as a shared partial image segment, use the following command sequence:

```
TASK   TASKID
START  (REG)XXXX
```

Where:

(REG)	is the segmentation register of the system .LIB partition.
XXXX	is the origin address of AIDS in the library segment.

The restart address of AIDS is (REG)XXXX+4. If AIDS is restarted at (REG)XXXX, another X'600' bytes of work storage are requested.

4.2.3 Using the START Command

The OS/32 START command is used to start AIDS and to provide arguments specifying alternate logical unit (lu) assignments in place of the expected default assignments.

DEFAULT ASSIGNMENTS	MEANING
2	Binary device
3	List device
5	Command input device

The following directive shows the key words that are recognized in the START command. All are optional and can be specified in any order, separated by commas.

Format:

$$\underline{\text{START}} \left[\begin{array}{l} \left[,\underline{\text{BINARY}}=\left[\text{lu}\left[/\text{dev}:\right]\right]\ \left[\text{fd}_1\right]\right] \\ \left[,\underline{\text{COMMAND}}=\left[\text{lu}\left[/\text{dev}:\right]\right]\ \left[\text{fd}_2\right]\right] \\ \left[,\underline{\text{LIST}}=\left[\text{lu}\left[/\text{dev}:\right]\right]\ \left[\text{fd}_3\right]\right] \end{array}\right]$$

Parameters:

BINARY	specifies that an optional lu and/or file or device is desired for any requested binary output.
COMMAND	specifies that an optional lu and/or file or device is desired for command response. All directives are read from this unit.
LIST	specifies that an optional lu and/or file or device is desired for the listing of printed output.

Examples:

Assign lu6 for binary output to the null device.

```
ST,BI=6/null:
```

Assign default list lu to a file called temp.

```
ST,LI=temp
```

Assign the command device to lul2, the list device to the printer and the default binary output to the null device.

```
ST,CO=12,LI=pr:,BI=null:
```

4.3 ERROR MESSAGES

The following error messages are printed by AIDS when error conditions are encountered:

ADDRESS OUTSIDE OF ALLOCATED MEMORY

An OPEN, REPLACE or DUMP command specifies a location outside of a user's segment.

BIAS ERR

A relative address computation yielded an invalid address.

BX REG ERR

The interpreter encountered a BXH and BXLE instruction with the R1 field greater than 13.

DATA ERR

Data or characters of other than the anticipated format are found in the directive field.

DUP ERR

An attempt was made to insert a sentinel that already exists.

EXCESS/DUPLICATE START OPTIONS

More than the required number of logical units are requested, or a previously requested unit is duplicated.

FILE DESCRIPTOR ERROR IN START OPTIONS

An error in the typing of the file descriptor (fd) is detected or a nonexistent file was specified.

FULL

An attempt was made to insert a sentinel in a list that already contains eight entries.

II to INTPRT XXXXXX

The AIDS interpreter encountered a privileged or illegal instruction in the execution thread. XXXXXX is the 6-digit hexadecimal location counter.

ILG OPT

An illegal option has been entered.

ILG SVC

An SVC14 has occurred that was not set by AIDS.

ILL CMD

The command key on the last directive was not valid.

ILLEGAL LU SELECTION

An lu assignment is requested with a negative value, or it exceeds 254 decimals.

ILLEGAL START OPTIONS

A misspelled or incorrect key word occurs in the START command.

INSUFFICIENT SPACE FOR GO OPTIONS

Not enough memory is available to store user options as input.

INVALID FLOATING POINT NUMBER

An attempt was made to display an invalid floating point number; i.e., a value that could not be normalized.

I/O ERR XXXX

An error was encountered during input/output (I/O). XXXX represents the information returned by the supervisor to the status word of the parameter block associated with the I/O.

LIM ERR

Limits on trace directive overlap an already existing trace region.

MOD ERR

Information required in the directive is missing.

NO SUCH

A register has been requested that does not exist.

OPT MSNG

An option is missing in the directive.

SNTX

The user has violated prescribed directive syntax.

OS/32 AIDS Rnn-nn

AIDS is initially entered and memory allocation is complete where n is the current revision level.

The disposition of all errors is that AIDS returns to the command mode and waits for the next directive. The exception is the batch mode, which causes AIDS to execute an SVC3 end of job (EOJ) to return to the OS.

APPENDIX A
OS/32 AIDS COMMAND SUMMARY

A [option] $data_1$ [,$data_2$]

B

BI [address]

C [option] data

D [option] $address_1$,$address_2$

EN

X [T] [count] [,address]

GO [address] [,user start options]

I { X [*] xaddress [*] [,delay]
  { P [H] [*] paddress [*]
  { T [B] [*] $taddress_1$ [*] ,$taddress_2$ [*]
  { S [option] [*] $address_1$ [*] ,$address_2$ [*] [,delay]

J [option]

L

LU U_1 ,U_2 ,U_3

N

NB

NL

O [option] address [,count]

{spacebar / CR} [option]

- [option]

P

R [option] data [,count]

S [option] $data_1$ [,$data_2$]

T{S / C}

W

Z{P / S / T / X / Z} [address]

APPENDIX B
SAMPLE DUMP FORMATS

```
H   H DUMP FROM 012344: TO 012384:
012344: -14272     32 -11703      0 -11767      4   4104 -11767
012354:      3   4104 -11767      2   4104 -11767      1 -11703
012364:      6 -14272     41 -11703      5    773 -14272     58
012374: -15408    256   9013  23343     56 -14272     82 -11703
012384:      6 -14208      6  16800   -132    773      0  -6112

F  F DUMP FROM 012344: TO 012384:
012344:  100911168    2150985      53769     266248
012354: -771162109  269013513     135176 -771162111
012364: -766967802 -935329751 -766967803   50710592
012374:    3851216   16786229 1529806904 -935329710
012384: -766967802 -931135482 1101070204   50659328

C  C DUMP FROM 012EB8: TO 012F00:
012EB8: 4241 4C52 4152 4E52 434C 5220 4F52 5852 BALRARNRCLR ORXR
012EC8: 4C52 4352 5352 4D48 5220 4448 5220 5352 LRCRSRMHR DHR SR
012ED8: 4C53 534C 4C53 4C50 5357 5220 4D52 4452 LSSLLSLPSWR MRDR
012EE8: 4C49 5320 4C43 5320 4149 5320 5349 5320 LIS LCS AIS SIS
012EF8: 4C45 5220 4345 5220 4145 5220 5345 5220 LER CER AER SER

E  E DUMP FROM 015000: TO 015040:
015000: 123,456     .4E-2     85.1001      .129E-5
015010: 456982     9,87001  14.3          189.6
015020: -21.5      -1867.89 .118114E-13 756.21
015030: 1289.5     -11       .345E-12     .216E-3
015040: .789002E-2 .789602  .3456        123.457

D  D DUMP FROM 0050000: TO 005040:
005000: .24074531026496D22  .14806597998898D22
005010: -.1924827394061D52  -.56326638318394D-46
005020: -.67489226182806D47 -40.156254907208
005030: -.96249409446709    -.13449104575638
005040: .384911100435209D31  .384911100433459D31

X  X DUMP FROM 012EB8: TO 012F00:
012EB8: 4241 4C52 4152 4E52 434C 5220 4F52 5852
012EC8: 4C52 4352 5352 4D48 5220 4448 5220 5352
012ED8: 4C53 534C 4C53 4C50 5357 5220 4D52 4452
012EE8: 4C49 5320 4C43 5320 4149 5320 5349 5320
012EF8: 4C45 5220 4345 5220 4145 5220 5345 5220
```

```
Y  Y DUMP FROM 012EB8: TO 12F00:
012EB8: 42414C52 41524E52 434C5220 4F525852
012EC8: 4C524352 53524D48 52204448 52205352
012ED8: 4C53534C 4C534C50 53575220 4D524452
012EE8: 4C495320 4C435320 41495320 53495320
012EF8: 4C455220 43455220 41455220 53455220

A  A DUMP FROM 012000: TO 012028:
012000: 4100 F5CA BAL  0,115CE
012004: C6D0 0010 OHI  D,10
012008: 4300 80C6 B    120D2
01200C: C540 002E CLHI 4,2E
012010: 2134      BNES 012018
012012: C6D0 0020 OHI  D,20
012016: 230C      BS   01202E
012018: C540 003A CLHI 4,3A
01201C: 2134      BNES 012024
01201E: C6D0 0040 OHI  D,40
012022: 2306      BS   01202E
012024: C540 002A CLHI 4,2A
012028: 4330 804E BE   1207A


U - DUMP OF SENTINELS SET
PROTECT: 020000:  GR(F)   FR(4)  015000:  FR(2)   FR(0)   GR(1)
TRACE  : 017000: 013000: 012A00: 012800: 012600: 012301: 012200:
BREAK  : 015020: 015010: 01500E: 01500A: 015008: 015004: 015000:
SNAP   : 015000: 015000: 015000: 017000: 017000: 012345: 012345:
```

APPENDIX C
SUPPLEMENT TO THE OS/32 AIDS USER GUIDE FOR CODE COMPATIBLE MACHINE (CCM) SUPPORT

C.1 INTRODUCTION

This appendix is a supplement to the OS/32 AIDS User Guide. It describes the enhancements to OS/32 AIDS that provide CCM support. It is now possible to debug user tasks (u-tasks) that contain a mixture of both Perkin-Elmer instructions and CCM instructions. There is full support of user supervisor call 14 (SVC14) and maintenance of all functions of task queue servicing for either instruction set.

For detailed information about the CCM instruction set, see the Perkin-Elmer Series 3200 Code Compatible Machine (CCM) User's Manual. Information pertaining to the use of the CCM to convert and reformat IBM® programs is contained in the Perkin-Elmer Series 3200 Code Compatible Machine (CCM) Emulator User's Manual.

C.2 GENERAL FEATURES

OS/32 AIDS performs checks on the operating system (OS) and the target machine to properly select code routines unique to a Model 7/32 or a Perkin-Elmer Series 3200 Processor, with or without CCM, u-tasks or executive tasks (e-tasks), and machines with or without floating point capabilities. In addition, it sets the task status word (TSW) to process SVC14 calls, arithmetic, memory access and illegal instruction faults. OS/32 AIDS tests the system for these items and makes the appropriate adjustments to registers and flags.

C.3 CODE COMPATIBLE MACHINE (CCM) SUPPORT FEATURES

Various enhancements to OS/32 AIDS have been developed to allow consistency with the use of the CCM instruction set. The user should be aware of certain features that are now operational due to these enhancements and should make adjustments accordingly.

IBM® is a registered trademark of International Business Machines Corporation

C.3.1 Boundary Alignments

Memory references fully support nonintegrally aligned boundaries. This is consistent with the needs of the CCM instruction set, which permits halfword and fullword data to be located on odd address locations. However, opcodes in either instruction set continue to require halfword alignment. All other memory references by AIDS permit halfwords with odd byte and odd halfword alignments.

C.3.2 Breakpoint Sentinel

The breakpoint sentinel is the illegal opcode X'FEED'. This is located at location X'000A' relative to AIDS. The breakpoint sentinel can be altered to any illegal opcode before AIDS is started. However, if Perkin-Elmer and CCM code are both present in the u-task, this sentinel must be illegal in both instruction sets. To allow the breakpoint to function properly, the u-task should not issue an SVC9, which would disable the illegal instruction trap.

C.3.3 Fault Traps

Exception-handling traps, also called fault traps, are considered to be owned by AIDS. These include the illegal instruction, arithmetic fault, memory access fault, and data format fault traps. New TSWs in the user-dedicated location (UDL) will be altered by AIDS before control is given to the u-task to enable the fault traps. If alterations are made to any of these enable flags or to any of the new location fields of these fault traps by the u-task, some of the functions performed by AIDS will be defeated. In addition, analysis of u-task logic may be difficult.

C.3.4 User Task Status Word (TSW) Trap Handling

During code execution of AIDS, location X'0008' defines user TSW trap bits that are enabled by user code. All task queue enable bits that have been set by the user remain in effect during AIDS and user code execution. The previously described fault traps are always forced each time control is given to user code.

C.3.5 Data Format Options

Format modifiers have been added to two of the data format options. These modifiers immediately follow the option character when used with the appropriate directives. Options that are affected are C for character mode and A for disassembly mode. The modifiers allow selection of the ASCII or EBCDIC character set and Perkin-Elmer or CCM instruction set interpretation.

See Chapter 2 for a discussion of data format modes with character mode C in Section 2.6 and disassembly mode A in Section 2.9.

C.3.5.1 Character Mode (C, CA, CE)

The character mode has additional modifiers to allow the user to choose the ASCII or EBCDIC character set. The modifiers are A for the ASCII character set and E for the EBCDIC character set. Thus, the ASCII character set can be specified using CA and the EBCDIC character set can be chosen by using CE.

If not specified, C defaults to the modifier last selected or to the mode of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set. At the time of the initial start-up of AIDS, ASCII is selected as the character set.

AIDS accepts a string of characters as input until either a carriage return or a semicolon (;) is encountered. Input is stored starting at the current open location as either ASCII or EBCDIC coded characters. The character set used is determined by the modifier that is in effect.

C.3.5.2 Disassembly Format (A, AP, AC)

Disassembly format has additional modifiers that allow the user to specify either the Perkin-Elmer instruction set or the CCM instruction set. The modifiers are P for the Perkin-Elmer instruction set and C for the CCM instruction set. Thus, the Perkin-Elmer instruction set can be selected using AP and the CCM character set can be specified with the use of AC.

If not specified, A defaults to the modifier last selected or to the mode of the last user instruction executed, whichever occurred last. At the time of the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.

The format of disassembly output has been expanded to include the effective address of the operands when displaying CCM instructions. This display is based on the last known value of the user's registers. Since CCM operations usually reference a base register to establish addressability, this information will be useful in determining where variables, constants and branch targets are really located. However, except when actually tracing user code, register values are not always known during disassembly output. This may lead to incorrect effective addresses at times. For opcodes of format SS, both effective addresses are presented in the same order as specified in the instruction.

C.3.6 Register Address Conventions

A register address field is defined as the hexadecimal number of a general purpose or floating point register preceded by a register type specification character. General purpose registers are denoted by G, single precision floating point registers by U, and double precision floating point registers by W. Each is followed by a single hexadecimal digit specifying the register number.

Examples:

ADDRESS FIELD	MEANING
GA	GPR A
G2	GPR 2
U8	Single precision register 8
UE	Single precision register E
WC	Double precision register C
W4	Double precision register 4

C.3.7 General Directive Guidelines

The command directive is expanded to support greater freedom of format, as read from the logical command unit 5 (lu5). Spaces are now optional and unlimited in number between fields of the command, except where they are required to avoid ambiguity. Thus, 'OY1000' and 'OY 1000' are both correct and equivalent. However, 'OA800' is not the same as 'O A800'. The option characters are not permitted any spaces between other option characters or the command character. They are scanned left to right until complete or until the first character that is not a valid option is found. Field separation is indicated with either a comma or spaces.

The command line supports multiple directives. The semicolon character (;) is used to separate the multiple directives. Thus, a directive is terminated by either a semicolon or the return key at the end of the command line. Two consecutive semicolons or a command composed of only spaces is considered an empty command and is ignored. However, a command line that is entirely empty is a special case and is treated as an open next (+) directive.

Most directives now support defaults for both the options and data fields. The defaults have been selected to decrease the number of keystrokes for the most frequently performed functions. They are also logically related to the uses of the respective commands. See specific directives in the following sections for details of these defaults.

C.3.8 OS/32 AIDS Commands

The functional description of all directives with new features are identified in the following sections. They have been arranged in alphabetical order. Syntactical requirements, usage notes and special considerations for specific directives are indicated. See Table C-1 for a list of the enhanced directives along with the minimum acceptable abbreviations and legal format option characters.

TABLE C-1 OS/32 AIDS COMMAND ENHANCEMENTS SUMMARY (CCM)

COMMAND	MNEMONIC	FORMAT OPTION CHARACTERS
CONVERT	C	C, CA, CE, D, E, F, Y
DUMP	D	A, AC, AP, B, C, CA, CE, D, E, F, H, L, U, X, Y
EXECUTE	X	T, V, P, C
GO	G	P, C
JUMP	J	A, AC, AP, C, CA, CE, D, E, F, H, X, Y
OPEN	O	A, AC, AP, C, CA, CE, D, E, F, H, X, Y
OPEN NEXT	+	A, AC, AP, C, CA, CE, D, E, F, H, X, Y
OPEN PREV	-	A, AC, AP, C, CA, CE, D, E, F, H, X, Y
REPLACE	R	C, CA, CE, D, E, F, H, X, Y

In general, the format option character is optional and defaults to the last usage of a format option for all directives.

The asterisk modifier (*), which appears in some of the following directive formats, results in the display of the open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix, when the asterisk modifier (*) is used. Otherwise, memory addresses are displayed as a 6-digit hexadecimal field with a colon (:) suffix.

```
 ---------------
|    CONVERT    |
 ---------------
```

C.3.8.1 CONVERT Command

This command converts data from one data format to another.

Format:

C [option] data

Parameters:

option — specifies the type of data format conversions to take place with the choice of one of the following parameters:

- CY — specifies that data is to be converted from character to hexadecimal format.
- CAY — specifies that data is to be converted from ASCII to hexadecimal format.
- CEY — specifies that data is to be converted from EBCDIC to hexadecimal format.
- YC — specifies that data is to be converted from hexadecimal to character format.
- YCA — specifies that data is to be converted from hexadecimal to ASCII character format.
- YCE — specifies that data is to be converted from hexadecimal to EBCDIC character format.
- FY — specifies that data is to be converted from fullword decimal to hexadecimal format.
- YF — specifies that data is to be converted from hexadecimal to fullword decimal format.
- EY — specifies that data is to be converted from single precision floating point to hexadecimal format.

YE specifies that data is to be converted from hexadecimal to single precision floating point format.

DY specifies that data is to be converted from double precision floating point to hexadecimal format.

YD specifies that data is to be converted from hexadecimal to double precision floating point format.

data specifies the field to be converted.

Functional Details:

To allow the user to convert data from one format to another, the following conversion operations can be used:

Character to hexadecimal	C ⟶ Y
ASCII to hexadecimal	CA ⟶ Y
EBCDIC to hexadecimal	CE ⟶ Y
Hexadecimal to character	Y ⟶ C
Hexadecimal to ASCII	Y ⟶ CA
Hexadecimal to EBCDIC	Y ⟶ CE
Decimal to hexadecimal	F ⟶ Y
Hexadecimal to decimal	Y ⟶ F
Single precision floating point to hexadecimal	E ⟶ Y
Hexadecimal to single precision floating point	Y ⟶ E
Double precision floating point to hexadecimal	D ⟶ Y
Hexadecimal to double precision floating point	Y ⟶ D

The execution of this command does not alter any memory or mode options within AIDS. Halfword modes are not applicable.

Examples:

Hexadecimal to character (defaults to option in effect):

```
>CYC 44
     D
```

Hexadecimal to ASCII:

```
>CYCA 44
      D
```

Hexadecimal to EBCDIC:

```
>CYCE C5
      E
```

ASCII to hexadecimal:

```
>CCAY D
      44
```

EBCDIC to hexadecimal:

```
>CCEY E
      C5
```

Decimal to hexadecimal:

```
>CFY 2858
     00000B2A
```

Single precision floating point to hexadecimal:

```
>CEY 10.0
     41A00000
```

C.3.8.2 DUMP Command

The DUMP directive outputs the contents of a section of memory or register space as defined by the two address operands or a start address and an item count. The format of the dump printout is specified by the format option in the directive.

Format:

$$D[option]\ [*]\ [address_1]\ \left[\left\{\begin{matrix},address_2\\ snnn\end{matrix}\right\}\right]$$

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output is to be displayed in assembly language format. It defaults to the instruction set of the option last selected (AC or AP) or the instruction set of the last user instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.
- AC — specifies that the printed output is to be displayed in assembly language format when using the CCM instruction set.
- AP — specifies that the printed output is to be displayed in assembly language format when using the Perkin-Elmer instruction set.
- B — specifies that the output is to be written to lu2 in memory image binary format.

C specifies that the printed output is to be displayed in character format. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.

CA specifies that the printed output is to be displayed in the ASCII character set.

CE specifies that the printed output is to be displayed in the EBCDIC character set.

D specifies that the printed output is to be displayed in double precision floating point format.

E specifies that the printed output is to be displayed in single precision floating point format.

F specifies that the printed output is to be displayed in fullword decimal format.

H specifies that the printed output is to be displayed in halfword decimal format.

L specifies that the output is to be written to lu2 in loader (binary object) format, (excutable form).

U specifies that the printed output is to be displayed as a listing of all breakpoint locations within the dump.

X specifies that the printed output is to be displayed in halfword hexadecimal format.

Y specifies that the printed output is to be displayed in fullword hexadecimal format.

* displays the open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix.

$address_1$ specifies the start address of the dump. If omitted, it defaults to the next address since the last dump. If the last dump was a register dump, the default is register 0.

$address_2$ specifies the end address of the dump. If omitted, it defaults to register 15 if the start address is a general register. Register 14 is the default end address if the start register is a floating point register. If start is a memory location, the default end address is start + 127. The default end register is start + 255 for loader or binary format. For disassembly format, the default end register is 10 opcodes.

snnn specifies a swath specification for the high limit where nnn is the decimal number of items to be displayed.

Functional Details:

The existence of active data formats causes erroneous dumps. Therefore, remove all data formats with the ZAP command before dumping.

The DUMP command displays the address locations of all set program sentinels. If a bias is set in AIDS, and the sentinel address location is to add the bias, the address locations are printed relative to the bias and indicated by the character R after the address location.

The output of all the dump parameters is written to lu3 except the loader format (L) and the memory image binary (B), which are written to lu2.

Items are defined as registers if the low limit is a register; otherwise they are defined as units of the format type.

The following is a list of the DUMP command data formats:

TYPE	OPTION
Disassembly	A
Disassembly (CCM instruction set)	AC
Disassembly (Perkin-Elmer instruction set)	AP
Memory image binary	B
Character	C
Character (ASCII character set)	CA
Character (EBCDIC character set)	CE
Double precision floating point	D
Single precision floating point	E
Fullword decimal	F
Halfword decimal	H
Loader	L
Listing of sentinel addresses	U
Halfword hexadecimal	X
Fullword hexadecimal	Y
Relative address format	*

Examples:

Display all the general registers in fullword hexadecimal format.

```
>DYG
```

Display all the double precision floating point registers in fullword hexadecimal format.

```
>DYW
```

Disassemble the next 10 opcodes.

```
>DA
```

Display the next 200 halfwords in hexadecimal format.

```
>DX,S200
```

C.3.8.3 END Command

This command terminates AIDS processing and returns control to the operating system with end of task code 0.

Format:

END

Functional Details:

EN or END will be the acceptable input for this command within the CCM version of AIDS. See Section 3.7 for the acceptable input in the alternate version of AIDS.

| EXECUTE |

C.3.8.4 EXECUTE Command

This directive causes one or more user program instruction to be executed interpretatively starting at a specified address.

Format:

$$X[T][V]\left[\begin{Bmatrix}P\\C\end{Bmatrix}\right][count]\ [,address]$$

Parameters:

T	specifies that each instruction and its address will be displayed to a print device upon execution.
V	specifies that interpretative execution and/or tracing of code is desired. Trace flow will include the SVC support code.
P	specifies that the Perkin-Elmer instruction set is to be used. If a particular instruction set mode, Perkin-Elmer (XP) or CCM (XC), is not specified, the directive defaults to the last selected mode or the mode of the last instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer mode is selected.
C	specifies that the CCM instruction set is to be used. If a particular instruction set mode, Perkin-Elmer (XP) or CCM (XC), is not specified, the directive defaults to the last selected mode or the mode of the last instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer mode is selected.
count	specifies the number of instructions to be executed. If this parameter is omitted, the default is 1.

address — is a 1- to 6-digit hexadecimal number specifying the starting address of the instructions to be executed. If this parameter is omitted and no starting address is given, execution begins with the next logical instruction following the last sentinel or interruption.

Functional Details:

After one or more program instructions are interpretatively executed, all user registers and the machine state are restored to interrupted or initial values, unless specifically altered by an AIDS directive. AIDS retains control of the central processing unit (CPU) resource by executing each user instruction interpretatively on behalf of the user program. After each instruction is executed, sentinel tables are searched for special handling.

The option V specifies that execution flow is to continue through a Perkin-Elmer SVC14 or a CCM SVC. If not specified, interpretative execution continues with the instruction following this SVC. The code resulting from the SVC14 TSW swap is not interpreted, thus treating this code in a similar manner to a code run by OS/32 on behalf of other Perkin-Elmer SVCs or CCM SVCPEs.

Note that a true breakpoint or fault trap within the SVC14 trap code causes a return to the AIDS command level regardless of the V option and the number of instructions to be executed. The instruction address location is printed relative to the bias if the address is greater than the bias.

Instructions in a shared task segment or in a write-protected partial image cannot be executed by this instruction.

Examples:

Execute and print four instructions starting at address location X'1A0' with no bias set.

```
>XT 4,1A0
```

Execute and print the next ten CCM instructions, with trace flow to include SVC support code.

```
>XVC 10
```

Execute and print four instructions starting at address location X'1A0' with no bias set.

```
>XT 4,1A0

 0001A0: 7820 FF74      LE      2,F74(F)
 0001A4: 3842           LER     4,2
 0001A6: 7F60 FF8E      SU      6,F8E(F)
 0001AA: 7E60 FFBA      AU      6,FBA(F)
```

Execute and print two instructions starting at address location X'1A0' with the bias set to X'100'.

```
>BI 100
>XT 2,1A0

 0000A0R 7820 FF74      LE      2,F74(F)
 0000A4R 3842           LER     4,2
```

Execute the next logical instruction without printing the instructions.

```
>X
```

Execute and print the next instruction with the bias set to X'100'.

```
>XT

 0000AAR 7E60 FFBA      AU      6,FBA(F)
```

NOTE

The AIDS location counter always points to the next logical executable instruction.

C.3.8.5 GO Command

This directive starts or continues the processing of a user program at a specified address. If no address is given, processing continues with the next logical instruction following the last sentinel or interruption.

Format:

GO[{P / C}] [address] [,user start options]

Parameters:

P	specifies that the Perkin-Elmer instruction set is being selected.
C	specifies that the CCM instruction set is being selected.
address	is a 1- to 6- digit hexadecimal number specifying the location at which execution of the GO command starts. If this parameter is omitted, execution continues with the next logical instruction following the last specified sentinel encountered (P, S, T, X). The default value of the address field is initially X'100'.
user start options	specifies any options the user program needs to execute correctly, such as logical units, files or devices. A maximum of eighty characters is allowed.

Functional Details:

All user registers, condition codes and the machine state are restored to their initial or interrupted values, unless specifically altered by AIDS directives.

If only an address field is specified, AIDS will place a carriage return at the current UTOP.

If user start options are specified, all characters from the one immediately following the comma through the carriage return (80 characters maximum) will be moved to the current UTOP.

If the GO command is a continuation following a breakpoint, the instruction at the breakpoint location is executed and normal execution proceeds. The breakpoint is not removed.

Examples:

Start program processing at address X'3000'.

```
>GO 3000
```

Start program processing of Perkin-Elmer instructions at current open location + 10.

```
>GP *+10
```

Continue executing from last breakpoint.

```
>GO
```

Execute program at address X'100' after placing options at UTOP for user.

```
>GO 100,IN=D0,OUT=D1,LIST=PR:
```

Start program processing of CCM instructions at address X'1000'.

```
>GC 1000
```

C.3.8.6 JUMP Command

This command displays and opens the location pointed to by the contents of the current open location.

Format:

J[option] [*]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output of the contents of the new current open location will be displayed in assembly language format. It defaults to AC or AP depending on the instruction set last selected or the instruction set of the last user instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.
- AC — specifies that the printed output of the contents of the new current open location will be displayed in assembly language format when using the CCM instruction set.
- AP — specifies that the printed output of the contents of the new current open location will be displayed in assembly language format when using the Perkin-Elmer instruction set.

C specifies that the printed output of the contents of the new current open location will be displayed in character format. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.

CA specifies that the printed output of the contents of the new current open location will be displayed in the ASCII character set.

CE specifies that the printed output of the contents of the new current open location will be displayed in the EBCDIC character set.

D specifies that the printed output of the contents of the new current open location will be displayed in double precision floating point format.

E specifies that the printed output of the contents of the new current open location will be displayed in single precision floating point format.

F specifies that the printed output of the contents of the new current open location will be displayed in fullword decimal format.

H specifies that the printed output of the contents of the new current open location will be displayed in halfword decimal format.

X specifies that the printed output of the contents of the new current open location will be displayed in halfword hexadecimal format.

Y specifies that the printed output of the contents of the new current open location will be displayed in fullword hexadecimal format.

*	displays the open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix.

Functional Details:

If no data format parameters are specified, the current open mode is used as the default. The new open location becomes the new current open location and the contents of the location pointed to by the current open location is interpreted as a fullword address.

Example:

Display the contents of the location pointed to by the contents of the current open location.

```
>OX  500
 000500:   3140
>JX
 003140:   1AE
```

| OPEN |

C.3.8.7 OPEN Command

This directive displays the contents of the memory location or register in the data format indicated by the format option character. A count specification allows for the display of multiple locations.

Format:

0[option][*] address [,count]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output will be displayed in assembly language format. It defaults to the instruction set of the option last selected (AC or AP) or the instruction set of the last user instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.
- AC — specifies that the printed output will be displayed in assembly language format when using the CCM instruction set.
- AP — specifies that the printed output will be displayed in assembly language format when using the Perkin-Elmer instruction set.
- C — specifies that the printed output will be displayed in character format. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.

CA specifies that the printed output will be displayed in the ASCII character set.

CE specifies that the printed output will be displayed in the EBCDIC character set.

D specifies that the printed output will be displayed in double precision floating point format.

E specifies that the printed output will be displayed in single precision floating point format.

F specifies that the printed output will be displayed in fullword decimal format.

H specifies that the printed output will be displayed in halfword decimal format.

X specifies that the printed output will be displayed in halfword hexadecimal format.

Y specifies that the printed output will be displayed in fullword hexadecimal format.

* displays the open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix.

address is a 1- to 6-digit hexadecimal number specifying a relocatable or absolute address. The default for address is the current open location.

count is a decimal number specifying the number of logical data locations or registers to be opened and displayed beyond the indicated address. The count includes the present location being opened. If this parameter is omitted, a default of one is assumed. The use of this parameter allows for the display of multiple locations.

Examples:

Open location 1004 and display in disassembly format for three consecutive locations.

```
>OA 1004,3

 001004:   1A22       AR    2,2
 001006:   43308062   IC    3,62(8)
 00100A:   48300045   LH    3,45
```

Open location 1006 and display in character format.

```
>OC 1006

 001006:  C0
```

Open double precision floating point register A and display in hexadecimal.

```
>OY WA

 DF (A) 43AC1000 76390000
```

Open single precision floating point register C and display in floating point.

```
>OE UC

 FP (C) 10
```

Open general purpose registers 5 through 7 and display in fullword decimal.

```
>OF G5,3

 GP(5)     13
 GP(6)     901437
 GP(7)     -213
```

Open location 21F0 + 10 in halfword hexadecimal with the current open location at address 0021F0.

```
>0 *+10

 002200:  FE91
```

Open relative location 1000 in character mode with the bias value set to 2000.

```
>OC 1000*

 003000:  EB
```

NOTE

An asterisk (*) specifies that the address field is relative. The character R is printed after the address location.

```
>OC* 1000*

001000R EB
```

| OPEN NEXT |

C.3.8.8 OPEN NEXT Command

This directive displays the next logical location or register in the format specified by the format option character. If an option is not specified, the mode last used by AIDS is used in the display.

Format:

+[option] [*]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output of the next logical location or register will be displayed in assembly language format. It defaults to AC or AP depending on the instruction set last selected or the instruction set of the last user instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.
- AC — specifies that the printed output of the next logical location or register will be displayed in assembly language format when using the CCM instruction set.
- AP — specifies that the printed output of the next logical location or register will be displayed in assembly language format when using the Perkin-Elmer instruction set.

C specifies that the printed output of the next logical location or register will be displayed in character format. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.

CA specifies that the printed output will be displayed in the ASCII character set.

CE specifies that the printed output will be displayed in the EBCDIC character set.

D specifies that the next logical data location or register will be displayed in double precision floating point format.

E specifies that the next logical data location or register will be displayed in single precision floating point format

F specifies that the next logical data location or register will be displayed in fullword decimal format.

H specifies that the next logical data location or register will be displayed in halfword decimal format.

X specifies that the next logical data location or register will be displayed in halfword hexadecimal format.

Y specifies that the next logical data location or register will be displayed in fullword hexadecimal format.

* displays open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix.

Functional Details:

A special form of this directive is the empty command line, which requests the opening of the next logical location or register in the same format as the current location display.

Example:

```
>OH 1000
 001000:   4120
>(Space)
 001002:    8830
>OF .5
 GP (5)     10
>Carriage return
 GP (6)     20141
>Carriage return
 GP (7)     0
```

C.3.8.9 OPEN PREVIOUS Command

This command displays the location or register immediately preceding the current open location or register in the format specified by the format option parameter.

Format:

```
-[option][*]
```

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- A — specifies that the printed output of the preceding location or register will be displayed in assembly language format. It defaults to AC or AP depending on the instruction set last selected or the instruction set of the last user instruction executed, whichever occurred last. At the initial start-up of AIDS, the Perkin-Elmer instruction set is in effect.
- AC — specifies that the printed output of the next logical location or register will be displayed in assembly language format when using the CCM instruction set.
- AP — specifies that the printed output of the preceding location or register will be displayed in assembly language format when using the Perkin-Elmer instruction set.

C specifies that the printed output of the preceding location or register will be displayed in character format. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.

CA specifies that the printed output will be displayed in the ASCII character set.

CE specifies that the printed output will be displayed in the EBCDIC character set.

D specifies that the preceding location or register will be displayed in double precision floating point format.

E specifies that the preceding location or register will be displayed in single precision floating point format.

F specifies that the preceding location or register will be displayed in fullword decimal format.

H specifies that the preceding location or register will be displayed in halfword decimal format.

X specifies that the preceding location or register will be displayed in halfword hexadecimal format.

Y specifies that the preceding location or register will be displayed in fullword hexadecimal format.

* displays the open cell address or location counter in relative address format. This is displayed as a 6-digit hexadecimal memory address with an R suffix.

Functional Details:

If no data format parameter is specified, the last one specified is used.

Example:

```
>OH 1000
 001000:    31F0
>-
 000FFE:    389A
>OF .7
 GP (7)     0
>-
 GP (6)     20141
>-
 GP (5)     10
```

C.3.8.10 REPLACE Command

This command replaces the contents of the current open location or register with new data specified in the operand field of the directive.

Format:

R[option] data[,count]

Parameters:

option — specifies the type of format in effect with the choice of one of the following parameters:

- C — specifies that the new contents are in either the ASCII or EBCDIC character set. It defaults to the character set of the option last selected (CA or CE) or the character set of the last user instruction executed, whichever occurred last. If the last user code executed was Perkin-Elmer, ASCII is the default character set. If the last user code executed was CCM, EBCDIC is the default character set.
- CA — specifies that the new contents are an ASCII character string.
- CE — specifies that the new contents are an EBCDIC character string.
- D — specifies that the new contents are set to double precision floating point format.
- E — specifies that the new contents are set to single precision floating point format.
- F — specifies that the new contents are a 1- to 10-digit fullword decimal number.
- H — specifies that the new contents are a 1- to 5-digit halfword decimal number.

X specifies that the new contents is a 1- to 4-digit hexadecimal number.

Y specifies that the new contents is a 1- to 8-digit hexadecimal number.

data specifies the data to be used as the new contents.

count specifies the number of locations or registers being replaced with new data. A contiguous block of memory can be initialized to a specific value through the use of the count option. The count value defaults to one.

Functional Details:

Execution of the REPLACE command automatically increments the current open location to the next logical location, i.e., the amount of memory operated on by the REPLACE command. This enables the user to conveniently replace memory without opening the next location.

Data in shared task segments or in write-protected partial image segments cannot be modified by this command.

Examples:

Set all double precision floating point registers to 0.

```
>OD W0          Sets the current open location to double
                precision floating point register (0).
 DF (0)         -40.156254907208
>R 0,8          Replaces the contents of eight consecutive
                registers with zero with the current open
                register, which is the double precision
                floating point register 0.
```

Set single precision floating point register A to 385.671

```
>OE UA          Opens single precision floating point register
                (A).
 FP (A)         10
>R  385.671     Replaces the contents of the single precision
                floating point register A with the value.
```

Set all fullwords in memory from location X'1000' to location X'2000' to a value of -1.

```
>OF 1000        Opens memory location X'1000'.
 001000:        13501
>R -1,2048      Replaces 2048 fullwords with the fullword
                decimal value -1.
```

Set the value of the current open location to halfword decimal 10353.

```
>RH 10353
```

Store the characters ABCD in the current open fullword or current open register.

```
>RC ABCD
```

NOTE

When replacing in character mode (ASCII or EBCDIC), do not overwrite memory beyond the desired location. AIDS accepts a string of up to 60 characters or up to a carriage return. Therefore, it is possible to enter a string larger than the anticipated space.

C.3.9 Limitations

AIDS does not recognize breakpoints that are targets of the CCM EX instruction. These breakpoints should be handled by setting the breakpoint on the EX operation and not on its target.

The EXECUTE command's V option is not currently implemented.

APPENDIX D
OS/32 AIDS GLOSSARY

The following terms are used widely in this document. Their meanings are established here for the reader.

Absolute	refers to a fixed memory address within the user program space.
AIDS	is the Automatic Interactive Debugging System.
Batch	refers to the mode of operation where the command input device is nonkeyboard (e.g., card reader). Batch differs from the interactive mode (keyboard) in command stream logging and error disposition.
Bias	is the value added to a relocatable address to obtain the true absolute location.
Cell	is a memory location operated upon by AIDS.
Delays	are event counters associated with breakpoint and snapshot dumps that cause sentinels to be encountered a controlled number of times before causing actual program interruption.
Displacement	is a value defined as the difference between an absolute address and the AIDS bias value.
Doubleword	is a 64-bit memory word that must begin on a doubleword boundary.
Environment	refers to the system under which AIDS is operating, as well as the particular peripheral and logical unit (lu) assignments at run-time.
Faults	are the arithmetic, machine malfunction and illegal instruction faults that must be considered under OS/32 to ensure proper operation of the AIDS interpreter.
Formats	are the available interpretations of binary data by AIDS.

Fullword	is a 32-bit memory unit that must begin on a 32-bit boundary.
Generalized AIDS Address	is an address form that the user employs to indicate: • Relative program address • Absolute program address • Local relative program addresses • General purpose registers • Single precision floating point registers • Double precision floating point registers
Halfword	is a 16-bit memory unit that must begin on a 16-bit boundary.
Log	is the recording of directives and error messages on the list unit.
Modifiers	are mandatory specification characters immediately following an AIDS command.
Operand Field	is a data field included in an AIDS directive.
Options	are nonmandatory command or address modifiers.
Program Space	is the user's partition.

INDEX

A

ADD command 3-2
Address conventions 1-3
Address fields
  absolute 1-3
  local relative 1-4
  registers 1-4
  relative 1-3
Addressability C-3
ASCII character set C-3
Asterisk 3-33
  modifier C-5

B

Base register C-3
BATCH command 3-4
Batch mode 1-2
  LOG command 3-26
BIAS command 3-5
Boundary alignments C-2
Branch trace. See Trace command.
Breakpoint command 3-17
Breakpoint sentinels C-2
Building AIDS. See establishment.

C

CCM
  enhancements C-1
  instruction set C-3
CCM enhanced commands
  CONVERT C-6
  DUMP C-9
  END C-13
  EXECUTE C-14
  GO C-17
  JUMP C-19
  OPEN C-22
  OPEN NEXT C-26
  OPEN PREVIOUS C-29
  REPLACE C-32
  summary C-5
CCM support features C-1
  boundary alignments C-2
  breakpoint sentinel C-2
  character mode C-3
  command line C-4
  commands C-5
  conversion operations C-7
  data format options C-2
  disassembly format C-3
  fault traps C-2
  limitations C-34
  register address conventions C-4
CCM support features (Continued)
  TSW trap handling C-2
Character
  format 2-2
  mode C-3
Character sets
  ASCII C-3
  default C-3
  EBCDIC C-3
Code compatible machine. See CCM.
Commands
  ADD 3-2
  BATCH 3-4
  BIAS 3-5
  Breakpoint 3-17
  CONVERT 3-6
  DUMP 3-9
  END 3-12
  ESTABLISH 4-3
  EXECUTE 3-13
  GO 3-15
  INCLUDE 4-2
  INSERT 3-17
  JUMP 3-24
  Link 4-2
  LOG 3-26
  LOGICAL UNIT 3-27
  NEXT 3-28
  NO BATCH 3-29
  NO LOG 3-30
  OPEN 3-31
  OPEN NEXT 3-34
  OPEN PREVIOUS 3-36
  PAUSE 3-38
  Protect 3-17
  REPLACE 3-39
  Snapshot dump 3-17
  START 4-4
  SUBTRACT 3-42
  syntax rules 3-1
  Trace 3-17
  TYPE 3-44
  WHERE 3-46
  ZAP 3-47
Commands summary A-1
Condition code 3-44
  restoration 3-15
CONTINUE command 3-38
Conversion operations 3-7
CONVERT command 3-6
  CCM enhanced C-6

D

Data format modes. See formats.
Data format options
  format modifiers C-2

Default lu assignments. See lu assignments.
Definitions D-1
Directives. See also commands.
  definition 1-3
  syntax 1-3
Disassembly format 2-4
  C-3
Double precision floating point format 2-3
DUMP command 3-9
  CCM data formats C-12
  CCM enhanced C-9
  data formats 3-11
Dump formats B-1
  sample B-1

E

EBCDIC character set C-3
END command 3-12
  CCM enhanced C-13
Error messages 4-5
  disposition 4-7
ESTABLISH command 4-3
Establishment
  into u-tasks 4-1
  shared partial segment 4-1
EXECUTE command 3-13
  CCM enhanced C-14

F

Fault traps C-2
Faults 4-1
Formats
  character 2-2
  disassembly 2-4
  double precision floating point 2-3
  fullword decimal 2-2
  fullword hexadecimal 2-1
  halfword decimal 2-2
  halfword hexadecimal 2-1
  single precision floating point 2-2
Fullword
  decimal format 2-2
  hexadecimal format 2-1

G

Glossary D-1
GO command 3-15
  CCM enhanced C-17

H

Halfword
  decimal format 2-2
  hexadecimal format 2-1

I

INCLUDE command 4-2
INSERT command
  Breakpoint 3-17
  Protect 3-17
  Snapshot dump 3-17
  Trace 3-17
Instruction sets
  CCM C-3
  Perkin-Elmer C-3
Interactive mode 1-2

J,K

JUMP command 3-24
  CCM enhanced C-19

L

Link commands 4-2
Location counter
  value 3-46
LOG command 3-26
LOGICAL UNIT command 3-27
Logical unit. See lu assignments.
lu assignments 3-27
  alternate 4-4
  binary output 1-6
  command input 1-6
  default 4-4
  list unit 1-6
  lu 1-6

M

Machine state
  restoration 3-15
Memory
  dump 3-9
  overwrite 3-40
  C-34
  replacement 3-40
Modes
  batch 1-2
  interactive 1-2

N

NEXT command 3-28
NO BATCH command 3-29
NO LOG command 3-30

O

OPEN command 3-31
  CCM enhanced C-22
OPEN NEXT command 3-34
  CCM enhanced C-26

OPEN PREVIOUS command 3-36
CCM enhanced C-29
Operating instructions 4-1
Operations 1-1
Operations, conversion 3-7
Origin address. See starting address.
OS commands
ASSIGN 3-27
CONTINUE 3-38
OS/32 AIDS
address conventions 1-3
capabilities 1-1
CCM support features C-1
command summary A-1
commands 3-1
directives 1-3
dump formats B-1
error messages 4-5
establishment 4-1
features 1-2
formats 2-1
glossary D-1
operating instructions 4-1
operations 1-1
requirements 1-1
start procedures 4-1
termination 3-12
OS/32 Link. See Link commands.
OS/32 Linkage Editor 4-2

P,Q

Partial image 4-2
PAUSE command 3-38
Perkin-Elmer instruction set C-3
Protect command 3-17

R

Register address
conventions C-4
field C-4
Registers
double precision floating point 1-4
general purpose 1-4
single precision floating point 1-4
REPLACE command 3-39
3-44
CCM enhanced C-32
Restart address 3-38
4-4

S

Sentinel
removal 3-47
Shared partial segment 4-2
multiple users 4-3
Single precision floating point format 2-2
Snapshot dump command 3-17
START command 4-4
Start procedures 4-1
Starting address 4-2
SUBTRACT command 3-42

T

Task segments 4-2
Task status word. See TSW.
Terminate OS/32 AIDS 3-12
Trace command 3-17
TSW 3-44
function 4-1
trap handling C-2
TYPE command 3-44

U,V

U-task
part of 4-2
UDLs 4-1
User-dedicated locations. See UDLs.
User task. See u-task.

W,X,Y

WHERE command 3-46

Z

ZAP command 3-47

Concurrent Computer Corporation

PUBLICATION COMMENT FORM

We try to make our publications easy to understand and free of errors. Our users are an integral source of information for improving future revisions. Please use this postage paid form to send us comments, corrections, suggestions, etc.

1. Publication number _____
2. Title of publication _____
3. Describe, providing page numbers, any technical errors you found. Attach additional sheet if necessary. _____
4. Was the publication easy to understand? If no, why not? _____
5. Were illustrations adequate? _____
6. What additions or deletions would you suggest? _____
7. Other comments: _____

From _____ Date _____

Position/Title _____

Company _____

Address _____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

Concurrent Computer Corporation
2 Crescent Place
Oceanport, NJ 07757

ATTN:
TECHNICAL SYSTEMS PUBLICATIONS DEPT.

FOLD FOLD

STAPLE STAPLE

9410