VSI Software

VSI OpenVMS

VSI DECset for OpenVMS Performance and Coverage Analyzer Reference Manual

VSI DECset for OpenVMS Performance and Coverage Analyzer Reference Manual:

VMS Software

Preface

v m s Software

This reference manual describes all the commands available with the Performance and Coverage Analyzer (PCA). The commands are listed in alphabetical order. Note that some commands are specific to the Collector or the Analyzer only, whereas others (labeled as PCA commands) are used by both.

For a tutorial description on the use of these commands, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

VSI seeks to continue the legendary development prowess and customer-first priorities that are so closely associated with the OpenVMS operating system and its original author, Digital Equipment Corporation.

2. Intended Audience

This reference is intended for experienced programmers and technical managers who are concerned with the performance of their application programs and will enter PCA commands from a command-line interface.

3. Document Structure

This reference contains the following information:

- A command dictionary that defines each command and provides an optional example

- A glossary of command parameters and qualifiers

- An error message list with suggestions for recovery procedures

4. Related Documents

The following documents are also helpful when using PCA:

- The VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide provides a tutorial description of the use of PCA commands from the command-line interface.

- The Guide to VSI DECset Performance and Coverage Analyzer for OpenVMS Systems provides a tutorial description of the use of PCA from the DECwindows interface, and contains other important user information.

- See Using DECset for OpenVMS Systems for information on using PCA with other DECset tools.

PCA is a component of the DECset tool kit. For more information on other DECset components, see the reference manuals for the individual components.

5. References to Other Products

Some older products that DECset components previously worked with might no longer be available or supported by VSI. Any reference in this manual to such products does not imply actual support, or that recent interoperability testing has been conducted with these products.

Note

These references serve only to provide examples to those who continue to use these products with DECset.

Refer to the Software Product Description for a current list of the products that the DECset components are warranted to interact with and support.

6. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: `<docinfo@vmssoftware.com>`. Users who have OpenVMS support contracts through VSI can contact `<support@vmssoftware.com>` for help with this product. Users who have OpenVMS support contracts through HPE should contact their HPE Support channel for assistance.

7. Conventions

VMScluster systems are now referred to as OpenVMS Cluster systems. Unless otherwise specified, references to OpenVMS Cluster systems or clusters in this document are synonymous with VMScluster systems.

The contents of the display examples for some utility commands described in this manual may differ slightly from the actual output provided by these commands on your system. However, when the behavior of a command differs significantly between OpenVMS Alpha and Integrity servers, that behavior is described in text and rendered, as appropriate, in separate examples.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Convention	Meaning
Ctrl/ *x*	A sequence such as **Ctrl**/ *x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
…	A horizontal ellipsis in examples indicates one of the following possibilities:

Convention	Meaning
	• Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
[|]	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are options; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold text	This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER= *name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Monospace type`	Monospace type indicates code examples and interactive screen displays. In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Chapter 1. Command Dictionary

1.1. Introduction

This dictionary lists all of the PCA commands alphabetically and provides a description and example of each command. Some commands are specific to the Collector or the Analyzer only, whereas others (labeled as PCA commands) are used by both. All commands apply to both the OpenVMS VAX and OpenVMS Alpha systems, unless otherwise indicated.

Following the alphabetical command list are two glossaries of related information:

• The Section 2.1 section provides a glossary of command parameters.

• The Section 2.2 section provides a glossary of command qualifiers.

The following tables list the commands grouped by the following categories:

• Table 1.1 lists the PCA commands.

• Table 1.2 lists the Analyzer commands.

• Table 1.3 lists the Collector commands.

Table 1.1. PCA Commands

@file-spec	SAVE	SHOW DATAFILE
ATTACH	SCROLL	SHOW DISPLAY
CANCEL	SEARCH	SHOW KEY
CANCEL COMMAND	SELECT	SHOW LANGUAGE
CANCEL DATAFILE	SET	SHOW LOG
CANCEL DISPLAY	SET DATAFILE	SHOW MAX_SOURCE_FILES
CANCEL LOG	SET DISPLAY	SHOW MODE
CANCEL SOURCE	SET KEY	SHOW SEARCH
CANCEL VERIFY	SET LANGUAGE	SHOW SELECT
CANCEL WINDOW	SET LOG	SHOW SOURCE
DEFINE/COMMAND	SET MAX_SOURCE_FILES	SHOW SYMBOL
DEFINE/KEY	SET MODE	SHOW TERMINAL
DELETE/COMMAND	SET PROMPT	SHOW TYPE
DELETE/KEY	SET SEARCH	SHOW VERIFY
DISPLAY	SET SOURCE	SHOW WINDOW
EXIT	SET TERMINAL	SPAWN
EXPAND	SET VERIFY	TYPE
EXTRACT	SET WINDOW	UNDEFINE/COMMAND
HELP	SHOW	UNDEFINE/KEY
MOVE	SHOW ALL	

Table 1.2. Analyzer Commands

APPEND	FIRST	SET PLOT
BACK	INCLUDE	SET TITLE
CANCEL ANC	LIST	SHOW ANC
CANCEL FILTER	MERGE	SHOW EDITOR
CANCEL PLOT	NEXT	SHOW FILTER
CANCEL TITLE	PAGE	SHOW PLOT
CURRENT	PLOT	SHOW RUN_DESCRIPTION
EDIT	PRINT	SHOW TITLE
EXCLUDE	SET ANC	TABULATE
FILE	SET EDITOR	
FIND	SET FILTER	

Table 1.3. Collector Commands

CANCEL ALL [1]	SET COUNTERS	SHOW COUNTERS
CANCEL COUNTERS	SET COVERAGE	SHOW COVERAGE
CANCEL COVERAGE	SET CPU_SAMPLING	SHOW CPU_SAMPLING
CANCEL CPU_SAMPLING	SET EVENT	SHOW EVENT
CANCEL EVENT	SET IO_SERVICES	SHOW IO_SERVICES
CANCEL IO_SERVICES	SET NOTE	SHOW NOTE
CANCEL PAGE_FAULTS	SET PAGE_FAULTS	SHOW PAGE_FAULTS
CANCEL PC_SAMPLING	SET PC_SAMPLING	SHOW PC_SAMPLING
CANCEL SERVICES	SET RUN_NAME	SHOW RUN_NAME
CANCEL STACK_PCS	SET SERVICES	SHOW SERVICES
CANCEL TASKING	SET STACK_PCS	SHOW STACK_PCS
CANCEL UNALIGNED [1]	SET TASKING	SHOW TASKING
CANCEL VCOUNTERS [2]	SET UNALIGNED [3]	SHOW UNALIGNED [3]
CANCEL VCPU_SAMPLING [2]	SET VCOUNTERS [2]	SHOW VCOUNTERS [2]
CANCEL VPC_SAMPLING [2]	SET VCPU_SAMPLING [2]	SHOW VCPU_SAMPLING [2]
GO	SET VPC_SAMPLING [2]	SHOW VPC_SAMPLING [2]

[1]This command cancels the data collection options established with the SET commands. Different SET commands apply to the OpenVMS VAX and OpenVMS Alpha systems.

[2]This command applies to the OpenVMS VAX system only.

[3]This command applies to the OpenVMS Alpha system only.

1.2. PCA Command Dictionary

@file-spec

@file-spec — Executes PCA commands contained in the specified command procedure or reads subsequent command input from the specified file or device.

Format

@file-spec

The @file-spec command reads and executes the commands contained in the specified command procedure as if the commands were entered at the terminal. The command procedure can contain any valid PCA commands, including other @file-spec commands.

When PCA executes an EXIT command in the command procedure or reaches the end of the file, it returns control to the command stream that invoked the command procedure. That command stream can be the terminal or a previous command procedure. Command procedure nesting is limited to 16 levels.

If you enter the SET VERIFY command, all commands read from the command procedure echo on the terminal.

Example

```
PCAC> @CMD_FILE.PCAC
```

This command reads Collector commands from the file CMD_FILE.PCAC until an EXIT command is encountered or the end-of-file is reached.

APPEND

APPEND — Appends the output from the most recent PLOT, TABULATE, or LIST command to an existing text file.

Format

APPEND [file-spec]

The APPEND Analyzer command concatenates many separate histograms, tables, or listings into a single file. You can specify a file, or accept the most recently created file as the default; however, the file must exist.

Example

```
PCAA> TABULATE/SOURCE PROGRAM BY LINE
PCAA> APPEND [MYDIR]FOOLINES.PCALIS
%PCA-I-APPFILE, appending to file DEV$:[MYDIR]FOOLINES.PCALIS
```

A table is generated with the TABULATE command. That table is then appended to the existing text file FOOLINES.PCALIS in directory [MYDIR].

ATTACH

ATTACH — Passes control of your terminal from your current process to another process.

Format

ATTACH process-name

The ATTACH command connects your terminal to another process. Using the ATTACH command, you can switch control back and forth between several different active processes. To create additional processes, use the SPAWN command from within PCA or at the DCL level.

Example

```
PCAC> SPAWN
            ! Spawn subprocess NICK_1
$ TYPE FOO.DAT
        ! Enter DCL commands in subprocess
text line in foo.dat
$ ATTACH NICK
        ! Attach to original process
%PCA-I-RETURNED, control returned to process NICK
PCAC> SHOW LANGUAGE
    ! Enter Collector commands
Language: PASCAL
PCAC> ATTACH NICK_1
    ! Attach back to subprocess
%PCA-I-ATTACHED, terminal now attached to process NICK_1
$
```

In this example, the Collector SPAWN command creates a subprocess. DCL commands are executed in the subprocess. A DCL ATTACH command reattaches the terminal to the Collector session. The Collector session continues with additional Collector commands until a Collector ATTACH command is entered to reconnect the terminal to the subprocess.

BACK

BACK — Traverses the current plot to display the previous leaf.

Format

BACK [subtree]

Command Qualifier	Default
/NOEXPAND	None

The BACK Analyzer command enables you to traverse a program's structure. The BACK command displays the current plot and positions an arrow to point to the bucket with the next highest number of data points (the previous leaf). The corresponding source lines for that portion of the program are displayed beside the plot. Other traverse commands are FIRST, NEXT, and CURRENT.

CANCEL

CANCEL — Cancels a variety of data collection options, definitions, and mode settings, depending on the keyword specified on the command line.

Format

CANCEL keyword [parameters]

CANCEL is not an individual command. Each CANCEL command consists of the word CANCEL followed by a required keyword. See the descriptions of the individual CANCEL commands for details.

CANCEL ALL

CANCEL ALL — Cancels all data collection options.

Format

CANCEL ALL

On OpenVMS Alpha and OpenVMS VAX systems, the CANCEL ALL Collector command cancels all data collection options established with the following SET commands:

SET COUNTERS	SET IO_SERVICES	SET STACK_PCS
SET COVERAGE	SET PAGE_FAULTS	SET TASKING
SET CPU_SAMPLING	SET PC_SAMPLING	
SET EVENT	SET SERVICES	

On OpenVMS Alpha systems only, the CANCEL ALL Collector command cancels all data collection options established with the following OpenVMS Alpha system-specific SET command:

SET UNALIGNED		

On OpenVMS VAX systems only, the CANCEL ALL Collector command cancels all data collection options established with the following OpenVMS VAX-specific SET commands:

SET VCOUNTERS	SET VCPU_SAMPLING	SET VPC_SAMPLING

In addition, the AST timer is reset to 10, the default.

CANCEL ANC

CANCEL ANC — Removes the ANC attribute from the specified coverage-set points.

Format

CANCEL ANC [nodespec]

Command Qualifier	Default
/ALL	None

CANCEL COUNTERS

CANCEL COUNTERS — Cancels the collection of exact execution counts at the specified program locations.

Format

CANCEL COUNTERS [nodespec, [nodespec...]]

Command Qualifier	Default
/ALL	None

The CANCEL COUNTERS Collector command cancels execution counters at the program locations specified by the node specification, or the /ALL qualifier. If you first set an execution counter at a program location with SET COUNTERS and then cancel it with CANCEL COUNTERS, no execution count is measured at that location when the GO command is entered.

Examples

1. `PCAC> CANCEL COUNTERS/ALL`

   This command cancels all previously established execution counters.

2. `PCAC> CANCEL COUNTERS PROGRAM BY ROUTINE`

   This command cancels the execution counters placed on every routine in the user program.

CANCEL COVERAGE

CANCEL COVERAGE — Cancels the collection of test coverage data at specified program locations.

Format

CANCEL COVERAGE [nodespec [,nodespec...]]

/ALL	None
/PREVIOUS	None

The CANCEL COVERAGE Collector command cancels the collection of test coverage data at the program locations specified by the node specification, or by the /ALL or /PREVIOUS qualifiers. If you first establish test coverage measurement at a program location with SET COVERAGE and then cancel it with CANCEL COVERAGE, no data is gathered at that location when the GO command is entered.

Examples

1. `PCAC> CANCEL COVERAGE/ALL`

   This command cancels all previously established test coverage breakpoints.

2. `PCAC> CANCEL COVERAGE PROGRAM BY CODEPATH`

   This command cancels the test coverage breakpoints on all code paths in the user program.

3. `PCAC> CANCEL COVERAGE/PREVIOUS`

This commands cancels the use of addresses from the previous runs.

CANCEL CPU_SAMPLING

CANCEL CPU_SAMPLING — Disables the collection of CPU sampling data.

Format

CANCEL CPU_SAMPLING

The CANCEL CPU_SAMPLING Collector command disables the collection of CPU sampling data established with an earlier SET CPU_SAMPLING command.

CANCEL DATAFILE

CANCEL DATAFILE — Cancels the performance data file established with the previous SET DATAFILE command.

Format

CANCEL DATAFILE

The CANCEL DATAFILE command closes the current performance data file. The Collector does not record data in that file when data collection starts. Instead, it uses the default data file, or the data file you establish with a subsequent SET DATAFILE command. The Analyzer does not read or process data until another SET DATAFILE command is entered.

CANCEL DISPLAY

CANCEL DISPLAY — Deletes the specified screen displays.

Format

CANCEL DISPLAY [display-name [,display-name...]]

Command Qualifier	Default
/ALL	None

The CANCEL DISPLAY command cancels the specified screen displays. When a screen display is canceled, it disappears from the screen and its contents and definition are permanently lost.

CANCEL EVENT

CANCEL EVENT — Cancels the collection of event markers established with previous SET EVENT commands.

Format

CANCEL EVENT [event-name [nodespec [,nodespec...]]]

Command Qualifier	Default
/ALL	None

The CANCEL EVENT Collector command cancels the collection of event markers for a specified event when the node specifications associated with that event are executed. Event markers for the *event-name* event are not recorded in the performance data file when these program locations are executed. If you specify an event name without a node specification, all node specifications associated with the event name are canceled.

Examples

1. PCAC> *CANCEL EVENT COMPUTE LINE %LINE 22*

    This command cancels the event marker established for event COMPUTE at line 22 of the program.

2. PCAC> *CANCEL EVENT TERM_IO*

    This command cancels all the event markers established for event TERM_IO.

CANCEL FILTER

CANCEL FILTER — Cancels filters established with previous SET FILTER commands.

Format

CANCEL FILTER [filter-name [,filter-name...]]

Command Qualifier	Default
/ALL	None

The CANCEL FILTER Analyzer command deletes the specified filters so they no longer filter the performance or coverage data used by the PLOT and TABULATE commands. Filters specify which data points from the performance data file to tally in histograms and tables.

CANCEL IO_SERVICES

CANCEL IO_SERVICES — Disables the collection of I/O services data.

Format

CANCEL IO_SERVICES

The CANCEL IO_SERVICES Collector command disables the collection of I/O services data established with an earlier SET IO_SERVICES command.

CANCEL LOG

CANCEL LOG — Disables the recording of PCA input and output in a log file.

Format

CANCEL LOG

The CANCEL LOG command disables the recording of input and output in a log file. This command cancels the effect of an earlier SET LOG command.

CANCEL PAGE_FAULTS

CANCEL PAGE_FAULTS — Disables the collection of page-fault data.

Format

CANCEL PAGE_FAULTS

The CANCEL PAGE_FAULTS Collector command disables the collection of page-fault data established with an earlier SET PAGE_FAULTS command.

CANCEL PC_SAMPLING

CANCEL PC_SAMPLING — Disables the collection of program counter sampling data.

Format

CANCEL PC_SAMPLING

The CANCEL PC_SAMPLING Collector command disables the collection of program counter sampling data established with an earlier SET PC_SAMPLING command.

CANCEL PLOT

CANCEL PLOT — Sets the SET PLOT defaults to their initial values.

Format

CANCEL PLOT

The CANCEL PLOT Analyzer command sets the SET PLOT default qualifiers to their initial values and sets the default node specification to PROGRAM_ADDRESS BY ROUTINE. The SET PLOT defaults are set with the SET PLOT command and specify qualifier defaults for the PLOT and TABULATE commands that have node specifications. (PLOT and TABULATE commands without node specifications take their defaults from the currently active plot.) The SET PLOT defaults also specify all qualifier and node specification defaults for PLOT and TABULATE commands that have the /DEFAULT qualifier.

The CANCEL PLOT command resets the default qualifiers to the following values:

/DESCENDING	/NOMAIN_IMAGE	/NOSTACK_DEPTH
/NOCHAIN_NAME	/NOMINIMUM	/NOWRAP
/NOCREATOR_PC	/NOPARENT_TASK	/PC_SAMPLING
/NOCUMULATIVE	/NOPATH_NAME	/SOURCE

/NOKEEP	/NOSCALE	/ZEROS

CANCEL SERVICES

CANCEL SERVICES — Disables the collection of system services data.

Format

CANCEL SERVICES

The CANCEL SERVICES Collector command disables the collection of system services data established with an earlier SET SERVICES command.

CANCEL SOURCE

CANCEL SOURCE — Cancels the current source file directory search list.

Format

CANCEL SOURCE

/EDIT	None
/MODULE=module-name	None

The CANCEL SOURCE command cancels the source-file directory search list established with a previous SET SOURCE command. After the directory search list is canceled, the Collector cannot find any source file that has been moved or copied since being compiled.

If you have established a directory search list for a specific module with the SET SOURCE/ MODULE=module-name command, you must use the same /MODULE qualifier on the CANCEL SOURCE command to cancel that search list.

If you have established a directory search list for the EDIT command using the SET SOURCE/EDIT command, you must use the same /EDIT qualifier on the CANCEL SOURCE command to cancel that search list.

CANCEL STACK_PCS

CANCEL STACK_PCS — Disables the collection of stack PC values (return addresses from the call stack) when other kinds of data are gathered.

Format

CANCEL STACK_PCS

The CANCEL STACK_PCS Collector command disables the collection of call stack PC values. It cancels the effect of an earlier SET STACK_PCS command.

CANCEL TASKING

CANCEL TASKING — Disables the collection of tasking data.

Format

CANCEL TASKING

The CANCEL TASKING Collector command disables the collection of tasking data established with a previous SET TASKING command.

CANCEL TITLE

CANCEL TITLE — Cancels the title text for histograms and tables established with a previous SET TITLE command.

Format

CANCEL TITLE

The CANCEL TITLE Analyzer command deletes the title text established with the previous SETTITLE command. Instead, the following default title text appears on each page of PLOT or TABULATE output:

```
        HP Performance and Coverage Analyzer
```

CANCEL UNALIGNED

CANCEL UNALIGNED — For OpenVMS Alpha systems only, the CANCEL UNALIGNED Collector command disables the collection of unaligned access fault data established with an earlier SET UNALIGNED command.

Format

CANCEL UNALIGNED

Example

```
PCAC> CANCEL UNALIGNED
```

This example disables the collection of unaligned access fault data.

CANCEL VCOUNTERS

CANCEL VCOUNTERS — On OpenVMS VAX systems only, cancels the collection of vector instruction counts at the specified program locations.

Format

CANCEL VCOUNTERS [nodespec, [nodespec...]]

Command Qualifier	Default
/ALL	None

The CANCEL VCOUNTERS Collector command cancels vector instruction counters at the program locations specified by the node specification, or by the /ALL qualifier. If you first set a vector

instruction counter at a program location with SET VCOUNTERS and then cancel it with CANCEL VCOUNTERS, no execution count is measured at that location when the GO command is entered.

Examples

1. PCAC> *CANCEL VCOUNTERS/ALL*

   This command cancels all previously established vector instruction counters.

2. PCAC> *CANCEL VCOUNTERS ROUTINE FOO BY VINSTRUCTION*

   This command cancels the vector instruction counters placed in routine FOO in the user program.

CANCEL VCPU_SAMPLING

CANCEL VCPU_SAMPLING — On OpenVMS VAX systems only, disables the collection of vector CPU sampling data.

Format

CANCEL VCPU_SAMPLING

The CANCEL VCPU_SAMPLING Collector command disables the collection of vector CPU sampling data established with an earlier SET VCPU_SAMPLING command.

CANCEL VERIFY

CANCEL VERIFY — Disables the echoing of command input to the terminal.

Format

CANCEL VERIFY

The CANCEL VERIFY command disables the echoing of command input to the terminal established with an earlier SET VERIFY command.

CANCEL VPC_SAMPLING

CANCEL VPC_SAMPLING — On OpenVMS VAX systems only, disables the collection of vector PC sampling data.

Format

CANCEL VPC_SAMPLING

The CANCEL VPC_SAMPLING Collector command disables the collection of vector PC sampling data established with an earlier SET VPC_SAMPLING command.

CANCEL WINDOW

CANCEL WINDOW — Deletes a screen window definition.

Format

CANCEL WINDOW [window-name [,window-name...]]

Command Qualifier	Default
/ALL	None

The CANCEL WINDOW command cancels the specified screen window definitions. After a window definition has been canceled, that window name can no longer be used on DISPLAY or SET DISPLAY commands. Only the definition of the window name is deleted; the actual screen region specified by the window name is not affected. Screen displays declared with the window name are not affected; however, the SHOW DISPLAY command shows numeric window coordinates for such windows instead of the canceled window name.

CURRENT

CURRENT — Displays the current traverse subtree. This is used within a series of traverse commands, essentially to refresh the screen.

Format

CURRENT [subtree]

Command Qualifier	Default
/NOEXPAND	None

The CURRENT Analyzer command allows you to traverse a program's structure. The CURRENT command displays the current plot with the pointer and corresponding source lines in the same position. The corresponding source lines for that portion of the program are displayed beside the plot.

The CURRENT SUBTREE command displays the current subtree at its first leaf (which is equivalent to the FIRST command). Other traverse commands are FIRST, NEXT, and BACK.

DEFINE/COMMAND

DEFINE/COMMAND — Associates a symbol name with a list of one or more PCA commands.

Format

DEFINE/COMMAND symbol-name = "command-string"

The DEFINE/COMMAND command associates a symbol name with a PCA command string. The command string consists of one or more PCA commands separated by semicolons, and must end with a partial or full PCA command. When the symbol name appears at the start of a subsequent PCA command, it is replaced by the command string. The resulting string is parsed and executed by PCA. DEFINE/COMMAND gives you the same command abbreviation capability as the symbol assignment statement (symbol-name := command-string) in DCL. The command string can contain previously defined commands when those symbols translate directly into PCA commands. (PCA supports one level of symbol translation.)

To display the current symbol definitions created by DEFINE/COMMAND, use the SHOW SYMBOL/DEFINED command.

Example

```
PCAC> DEFINE/COMMAND SA = "SHOW ALL"
PCAC> SA
```

This command associates the symbol SA with the PCA command SHOW ALL. When the SA command is used, SA is expanded to the equivalent command string, which PCA then executes.

DEFINE/KEY

DEFINE/KEY — Associates a PCA command string and a set of attributes with a key on the terminal keyboard.

Format

DEFINE/KEY key-name "command-string"

Command Qualifiers	Defaults
/[NO]ECHO	/ECHO
/[NO]LOCK_STATE	/NOLOCK_STATE
/[NO]LOG	/LOG
/[NO]TERMINATE	/NOTERMINATE

The DEFINE/KEY command assigns PCA command strings to keypad keys. When you press the appropriate keypad key, PCA enters the associated command string into your command line.

The key definition remains in effect until you redefine the key (by entering another DEFINE/KEY command for the same key), enter the DELETE/KEY command for that key, or exit. You can include key definitions in a command procedure, such as your PCA initialization file.

The *key-name* parameter specifies the key you want to define. Table 1.4 lists the standard key names.

Table 1.4. Standard Key Names

Key Name	LK201	VT100-series	VT52
PF1	PF1	PF1	[blue]
PF2	PF2	PF2	[red]
PF3	PF3	PF3	[gray]
PF4	PF4	PF4	- -
KP0, KP1, . . . , KP9	0, 1, . . . , 9	0, 1, . . . , 9	0, 1, . . . , 9
Period	.	.	.
Comma	,	,	n/a
Minus	-	-	n/a
Enter	Enter	ENTER	ENTER
Left	←	←	←

Key Name	LK201	VT100-series	VT52
Right	→	→	→
Find (E1)	Find	—	—
Insert Here (E2)	Insert Here	—	—
Remove (E3)	Remove	—	—
Select (E4)	Select	—	—
Prev Screen (E5)	Prev Screen	—	—
Next Screen (E6)	Next Screen	—	—
Help	Help	—	—
DO	Do	—	—
F6, F7, . . . , F20	F6, F7, . . . , F20	—	—

Example

```
PCAC> DEFINE/KEY/TERMINATE KP7 "SHOW ALL/FULL"
```

This command associates KP7 with the SHOW ALL/FULL command. Because the /TERMINATE qualifier is used, you do not need to press the Return key after pressing KP7 to execute the command.

DELETE/COMMAND

DELETE/COMMAND — Deletes the specified command definitions.

Format

DELETE/COMMAND symbol-name [, symbol-name...]

Example

```
PCAC> DELETE/COMMAND FOO, FEE
```

This example deletes the command definitions for symbols FOO and FEE.

DELETE/KEY

DELETE/KEY — Deletes key definitions established with the DEFINE/KEY command.

Format

DELETE/KEY key-name

/ALL	None
/[NO]LOG	/LOG

Example

```
PCAC> DELETE/KEY KP7
```

This command deletes the key definitions for KP7.

DISPLAY

DISPLAY — Outputs the specified displays on the screen or changes the attributes of those screen displays. This is also used to refresh the screen.

Format

DISPLAY [display-name [AT window-spec] [display-kind]

[, display-name [AT window-spec] [display-kind]...]]

DISPLAY/REFRESH

/CLEAR	None
/[NO]DYNAMIC	/DYNAMIC
/HIDE	None
/[NO]POP	/POP
/[NO]PUSH	/NOPUSH
/REFRESH	None
/REMOVE	None
/SIZE:n	None

The DISPLAY command shows the contents of the specified screen displays. Each specified display is placed on top of any other displays that occupy overlapping windows on the screen.

You can also use the DISPLAY command to change the attributes of the specified screen displays. You can create displays that do not appear on the screen, change their sizes, clear their contents, and change their window locations and display kinds.

Examples

1. PCAA> *DISPLAY PLOT*

   This command displays the screen display named PLOT. PLOT is placed over any other displays with overlapping windows.

2. PCAA> *DISPLAY/SIZE:200 OUT*

   This command changes the size of the OUT display so that OUT holds the 200 most recent lines of output instead of the default 100 lines. It also places OUT on top of any other displays with overlapping windows.

EDIT

EDIT — Enables you to use an editor (such as the VSI Language-Sensitive Editor) to edit the source file displayed by the current PLOT or TABULATE command.

Format

EDIT [[module-name \] line-number]

Command Qualifier	Default
/[NO]EXIT	/NOEXIT

The EDIT Analyzer command invokes an editor to edit the source file displayed by the current PLOT or TABULATE command. To get a source display, use the /SOURCE qualifier on the PLOT or TABULATE command and use a BY LINE or BY CODEPATH node specification.

If you do not specify a parameter on the EDIT command, the editor is positioned at the central line of the page currently shown on your terminal. The *module-name* and *line-number* parameters enable you to position the editor at a different source file or line.

When you exit from the editor, you are returned to the Analyzer session. However, if you use the /EXIT qualifier on the EDIT command, the Analyzer session is terminated before the editor is invoked.

The editor you call must be accessible from DCL level, and can be set with the SET EDITOR command. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

EXCLUDE

EXCLUDE — Excludes one datakind from a multiple datakind plot or table.

Format

EXCLUDE n

The EXCLUDE Analyzer command removes one datakind from a multiple datakind plot. Multiple datakind plots are created by the INCLUDE command. The EXCLUDE command provides an incremental approach to building a plot or table. After you execute this command, a new plot or table is created and displayed. For complete information about multiple datakinds, see the Guide to VSI DECset Performance and Coverage Analyzer for OpenVMS Systems.

Example

```
PCAA> EXCLUDE 6
```

This command creates, then displays, a new plot or table that excludes the sixth datakind.

EXIT

EXIT — Ends the current session, or ends the execution of commands in a command procedure.

Format

EXIT

When you enter the EXIT command at the terminal, the session ends and you return to DCL level. You can also press Ctrl/Z to end the session.

When PCA executes an EXIT command in a command procedure, control returns to the command stream that invoked the command procedure. This command stream can be the terminal or a previous

command procedure. To return to the DCL command level after invoking PCA in a command procedure, define the logical name PCAA$INPUT to be a file that contains an EXIT command. PCA executes the PCAA$INIT command file at invocation and then gets its input from the PCAA$INPUT file. Because the PCAA$INPUT file has an EXIT command, you will then exit PCA.

EXPAND

EXPAND — Expands or contracts the window associated with a screen display.

Format

EXPAND [disp-name[, . . .]]

/DOWN[:n]	None
/LEFT[:n]	None
/RIGHT[:n]	None
/UP[:n]	None

The EXPAND command moves one or more display-window borders according to the qualifiers specified. You must specify at least one qualifier.

The EXPAND command does not affect the order of a display on the display pasteboard. Depending on the relative order of displays, the EXPAND command might cause the specified display to hide or uncover another display, or be hidden by another display, partially or totally.

Except for the PROMPT display, any display can be contracted to the point where it disappears (at which point it is marked as removed). It can then be expanded from that point. Contracting a display to the point where it disappears will cause it to lose any attributes that were selected for it. The PROMPT display cannot be contracted or expanded horizontally, but can be contracted vertically to a height of two lines.

A window border can be expanded only up to the edge of the screen. The left and top window borders cannot be expanded beyond the left and top edges of the display, respectively. The right border can be expanded up to 255 columns from the left display edge. The bottom border of a display can be expanded down only to the bottom edge of the display (to the end of the source module or routine's instructions).

Examples

1. PCAA> **EXPAND/RIGHT:6**

   The EXPAND command moves the right border of the current scrolling display to the right by 6 columns.

2. PCAA> **EXPAND/UP/RIGHT:-12 OUT2**

   The EXPAND command moves the top border of display OUT2 up by 1 line, and the right border to the left by 12 columns.

3. PCAA> **EXPAND/DOWN:99 SRC**

   The EXPAND command moves the bottom border of display SRC down to the bottom edge of the screen.

EXTRACT

EXTRACT — Saves the contents of a screen display in a file, or creates a file with all the commands necessary to create the current screen state at a later time.

Format

EXTRACT [disp-name[, . . .]] [file-spec]

/ALL	None
/APPEND	None
/SCREEN_LAYOUT	None

When you use the EXTRACT command to save the contents of a display into a file, only those lines that are currently stored in the display's memory buffer (as determined by the /SIZE qualifier on the DISPLAY or SET DISPLAY command) are written to the file.

You cannot extract the contents of the PROMPT display into a file.

Examples

1. `PCAA>` ***EXTRACT SRC***

   This command writes all the lines in display SRC into file PCA.TXT.

2. `PCAA>` ***EXTRACT/APPEND OUT [STEVE.WORK]MYFILE***

   This command appends all the lines in display OUT to the end of file [STEVE.WORK]MYFILE.TXT.

3. `PCAA>` ***EXTRACT/SCREEN_LAYOUT***

   This command writes the PCA commands needed to reconstruct the screen into file PCA.PCAA.

FILE

FILE — Writes the output from the most recent PLOT, TABULATE, INCLUDE, EXCLUDE, LIST, or traverse command to a text file.

Format

FILE file-spec

/APPEND	None
/DDIF	None

The FILE Analyzer command writes the output to a file. To write a histogram, table, or listing of raw performance data to a file, enter a command to generate the report on the terminal, then enter a FILE command to write that report to a file. The text file can be printed or used as input to other programs.

When writing histograms and tables to a file, the FILE command assumes that there are 66 lines per page and formats the output accordingly. You can change the number of lines per page by defining the logical name SYS$LP_LINES at DCL level.

You can concatenate many separate histograms, tables, or listings into a single file by using the /APPEND qualifier or APPEND command.

Examples

1. `PCAA> ` **`PLOT/IO_SERVICES MODULE FOO BY LINE`**
    `PCAA> ` **`FILE FOOLINES`**
    `%PCA-I-CREFILE, creating file DEV$:[MYDIR]FOOLINES.PCALIS`

    A histogram is generated with the PLOT command. That histogram is then written to the text file FOOLINES.PCALIS.

2. `PCAA> ` **`LIST ALL`**
    `PCAA> ` **`FILE PING.DAT`**
    `%PCA-I-CREFILE, creating file DEV$:[MYDIR]PING.DAT`

    A listing of all raw data in the performance data file is generated with the LIST command. (If the output is long, it can be stopped with Ctrl/C.) That listing is written to the file PING.DAT. This file can later be printed or used as input to a data reduction program of your own.

3. `PCAA> ` **`FILE/DDIF HISTOGRAM.DDIF`**

    This example causes the display to be stored in DDIF format in the file HISTOGRAM.DDIF.

FIND

FIND — Finds and displays the page of PLOT or TABULATE output that contains the histogram bar or table entry for a specified label or line number.

Format

FIND [label-name]

The FIND Analyzer command finds the first page of histogram or table output that contains a specified label and displays that page on the terminal. It is useful for quickly finding specific bucket names in long histograms or tables.

Examples

1. `PCAA> ` **`PLOT PROGRAM BY ROUTINE`**
    `PCAA> ` **`FIND OUTPUT_TO_DATAFILE`**

    A histogram is generated by the PLOT command. Each bucket in the histogram corresponds to one routine in the program. The FIND command displays the histogram page containing the histogram bar for routine OUTPUT_TO_DATAFILE.

2. `PCAA> ` **`PLOT PROGRAM BY ROUTINE`**
    `PCAA> ` **`FIND PRIM*`**
    `PCAA> ` **`FIND`**

    A histogram is generated by the PLOT command. The FIND command displays the first histogram page containing a routine name beginning with PRIM. The second FIND command finds the next occurrence of such a routine name.

3. `PCAA> ` **`TABULATE ROUTINE PRIM_NUM_GEN BY LINE`**
    `PCAA> ` **`FIND %LINE 1024`**

A table is generated by the TABULATE command. The page containing the table entry for line 1024 is displayed by the FIND command.

FIRST

FIRST — Positions the current traverse subtree to display its first leaf.

Format

FIRST [subtree]

Command Qualifier	Default
/NOEXPAND	None

The FIRST Analyzer command allows you to traverse a program's structure. This command displays the current traverse subtree and positions an arrow to point to the bucket with the most data points. The corresponding source lines for that portion of the program are displayed beside the plot. Other traverse commands are NEXT, BACK, and CURRENT.

GO

GO — Executes a program and starts data collection.

Format

GO

Command Qualifier	Default
/NOCOLLECT	None

The GO Collector command executes a program and collects performance or coverage data while the program runs to completion. After entering the GO command, you cannot enter more Collector commands.

If you do not establish a performance data file with the SET DATAFILE command before entering the GO command, the Collector uses a default data file. The default data file has the same file name as your program's executable image file and a file type of .PCA. If you do not establish any data collection options before entering the GO command, the Collector gathers program counter sampling data and stack data by default.

HELP

HELP — Displays information about PCA commands and features. This information describes each command, its format, qualifiers, and parameters.

Format

HELP [topic [subtopic...]]

If you want information about a particular qualifier or parameter, specify it as a subtopic. If you want information about all command qualifiers, specify "Qualifier" as a subtopic. If you want information

about all parameters, specify "Parameter" as a subtopic. If you want all available information about a command, specify an asterisk (*) as a subtopic. If you want to redisplay the last help entry you viewed, enter a question mark (?).

INCLUDE

INCLUDE — Adds a new datakind to the plot. This command enables you to compare more than one datakind in the same collection.

Format

INCLUDE

/ANC	Defaults are taken from the current
/COUNTERS	active plot
/[NON]COVERAGE	
/CPU_SAMPLING	
/[NO]CREATOR_PC	
/[NO]CUMULATIVE[= *n*]	
/DEFAULT	
/EVENT	
/FAULT_ADDRESS	
/FILL=	
/IO_SERVICES	
/[NO]MAIN_IMAGE[=prog-unit]	
/PAGE_FAULTS	
/[NO]PARENT_TASK	
/PC_SAMPLING	
/PHYSICAL_IO_COUNT	
/POSITION= *n*	
/READ_COUNT	
/SERVICES	
/[NO]STACK_DEPTH= *n*	
/TASK_SWITCH	
/UNALIGNED	
/WRITE_COUNT	
/VCOUNTERS	
/VCPU_SAMPLING	
/VPC_SAMPLING	

The INCLUDE Analyzer command adds a new datakind into the current plot. It is used with the EXCLUDE command to incrementally build one plot that contains a maximum of eight datakinds per bucket. The datakinds must be compatible with the node specifications for the current plot. If you want a new set or order of the buckets, you must enter a new PLOT or TABULATE command. The

use of INCLUDE command qualifiers does not affect the old qualifiers in the PLOT/TABULATE default entry of the current active plot. If no active plot is available, an error message is displayed. You can apply a different filter to the datakind for every INCLUDE command. The first page of the newly created plot is displayed after you enter the command.

For more information on multiple datakinds, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Example

```
PCAA> PLOT/MAIN_IMAGE/IO_SERVICES/FILL=("io","pc") PROGRAM BY ROUTINE
PCAA> INCLUDE/PC_SAMPLING
PCAA> INCLUDE/PHYSICAL_IO/FILL="phys_io"
```

This series of commands plots I/O services, PC sampling, and physical I/O data points in the same histogram. The following example shows that the time is being spent doing physical I/O in the program's sorting routine:

```
          HP Performance and Coverage Analyzer                Page 1
      I/O System Service Calls (316152 data points total) - "io"
    Program Counter Sampling Data (1019612 data points total) - "pc"
       Total Physical I/O Counts (206391 data points total) - "phys_io"
Bucket Name            +----+----+----+----+----+----+----+----+----+----+
INPUT_DATA\            |
  INPUT_DATA . . . .   |ioioioioioioioioioioioioioioioioioioioioiioioio    79.1%
                       |pcpcpcp                                            11.9%
                       |p                                                   0.1%
SORT_PRIMES\           |
  SORT_PRIMES  . . .   |ioioioioi                                          13.9%
                       |pcpcpcpcpcpcpcpcpcpcpcpcpcpcpcpcpcpcpc             83.7%
                       |phys_iophys_iophys_iophys_iophys_iophys_iophy      99.8%
OUTPUT_PRIMES\         |
  OUTPUT_PRIMES   . .  |ioio                                                7.0%
                       |pc                                                  3.1%
                       |                                                    0.0%
PCA$PRIMES_1\          |
  PCA$PRIMES_1 . . .   |                                                    0.0%
                       +----+----+----+----+----+----+----+----+----+----+
```

LIST

LIST — Displays raw performance or coverage data. This command also displays other information stored in the performance data file.

Format

LIST datakind [,datakind...]

/CODEPATHS	None
/COVERAGE_SET	None
/CPT	None
/DMT	None
/DST	None

/EVENT_NAMES	None
/FILE_NAMES	None
/GST	None
/HEADER	None
/HEXADECIMAL	None
/IMAGE	None
/LOST_DATA=ALL	None
/LOST_DATA=PCA	None
/LOST_DATA=USER	None
/LOST_DATA=NONE	/LOST_DATA=NONE (eq. to / NOLOST_DATA)
/MODULE_NAMES	None
/NOLOST_DATA	/NOLOST_DATA (eq. to / LOST_DATA=NONE)
/SEQUENCE_NUMBERS	/NOSEQUENCE_NUMBERS
/SHAREABLE_NAMES	None
/START_TIME	/NOSTART_TIME
/TASK_NAME	None
/TASK_TYPE	None
/VCOUNTERS	None
/VCPU_SAMPLING	None
/VPC_SAMPLING	None

The LIST Analyzer command displays information stored in the performance data file. If you specify datakind parameters, it displays all corresponding performance or coverage data records from the file. Thus, you can inspect the raw performance or coverage data gathered by the Collector. If you specify the appropriate qualifiers, the LIST command displays other information stored in the performance data file (see the qualifier list). You must request at least one kind of information from the LIST command.

PCA can display extensive output from the LIST command. Stop the LIST output by pressing Ctrl/C.

The LIST output can be sent to the printer with the PRINT command, or written to a file with the FILE command. If you write the LIST output to a file, you can use it as input to your own data reduction programs.

Examples

1. PCAA> **LIST ALL**

   This command lists all performance or coverage data records in the data file.

2. PCAA> **LIST IO_SERVICE, PAGE_FAULTS**

   This command lists only I/O services and page fault data records from the data file.

3. PCAA> **LIST VCOUNTERS**

This command lists the vector instruction counter data from the data file.

4. `PCAA> ` **`LIST/HEADER`**

This command lists the header of the performance data file.

MERGE

MERGE — Combines several PCA performance data files into one file, or combines data extracted from several Software Performance Measurement (SPM) data files into one PCA data file.

Format

MERGE input-file-spec[,input-file-spec...]

/ANC	None
/BEFORE="time-spec"	None
/SINCE="time-spec"	None
/SPM_PROCESS_ID	None

The MERGE command makes it possible for you to combine several PCA performance data files, or to combine data extracted from several SPM data files into one PCA data file. This enables you to process and display the PC sampling data collected by SPM. Merged data is always appended to the currently opened performance data file. For more information on the MERGE command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

MOVE

MOVE — Moves a screen display vertically or horizontally across the screen.

Format

MOVE [disp-name[, . . .]]

/DOWN[:n]	None
/LEFT[:n]	None
/RIGHT[:n]	None
/UP[:n]	None

For each display specified, the MOVE command creates a window of the same dimensions elsewhere on the screen and maps the display to it, while maintaining the relative position of the text within the window. You must specify at least one qualifier.

The MOVE command does not change the order of a display on the display pasteboard. The MOVE command might cause the display to hide or uncover another display, or be hidden by another display (partially or totally). A display can be moved only up to the edge of the screen.

Examples

1. `PCAA> ` **`MOVE/LEFT`**

The MOVE command moves the current scrolling display to the left by one column.

2. `PCAA> MOVE/UP:3/RIGHT:5 NEW_OUT`

The MOVE command moves display NEW_OUT up by three lines and to the right by five columns.

NEXT

NEXT — Traverses the current plot to display the next most significant leaf.

Format

NEXT [subtree]

Command Qualifier	Default
/NOEXPAND	None

The NEXT Analyzer command enables you to traverse a program's structure. The initial NEXT command displays the first traverse subtree and positions an arrow to point to the bucket with the most data points. The corresponding source lines for that portion of the program are displayed beside the plot. Subsequent NEXT commands move the arrow to the bucket with the next most data points (the next leaf), and so on. Other traverse commands are FIRST, BACK, and CURRENT.

PAGE

PAGE — Displays a specified page of output on the terminal.

Format

PAGE parameter

The PAGE Analyzer command gives you random access to the pages of the histogram or table generated by the last PLOT, TABULATE, INCLUDE, EXCLUDE, or traverse command you entered. The PAGE command displays the next, current, previous, or first summary page, or a page with a specific page number. You cannot use the PAGE command unless there is a currently active plot or table.

Examples

1. `PCAA> PLOT/SOURCE PROGRAM BY LINE`
   `PCAA> PAGE 16`

   The PLOT command generates a histogram and displays page 1 of that histogram. The PAGE command then displays page 16 of the same histogram.

2. `PCAA> PAGE PREVIOUS`

   This command displays the previous page of the currently active plot. If the current page is page 16, this command displays page 15.

3. `PCAA> ` ***PAGE SUMMARY***

This command displays the first summary page of the currently active plot.

PLOT

PLOT — Generates performance or coverage histograms.

Format

PLOT [nodespec [,nodespec...]]

datakind qualifiers:	/PC_SAMPLING
/ANC	
/COUNTERS	
/COVERAGE	
/CPU_SAMPLING	
/EVENT	
/FAULT_ADDRESS	
/IO_SERVICES	
/NONCOVERAGE	
/PAGE_FAULTS	
/PC_SAMPLING	
/PHYSICAL_IO_COUNT	
/READ_COUNT	
/SERVICES	
/UNALIGNED	
/VCOUNTERS	
/VCPU_SAMPLING	
/VPC_SAMPLING	
/WRITE_COUNT	
Sorting qualifiers:	/DESCENDING
/ALPHABETICALLY[=[n:]m]	
/ASCENDING[=[n:]m]	
/DESCENDING[=[n:]m]	
/NOSORT[=[n:]m]	
Bucket selection qualifiers:	
/[NO]MAXIMUM[=n]	/NOMAXIMUM
/[NO]MINIMUM[=n]	/NOMINIMUM
/[NO]ZEROS	/ZEROS
Source-code display qualifiers:	

/[NO]SOURCE	/SOURCE
Program address selection qualifiers:	
/[NO]CUMULATIVE	/NOCUMULATIVE
/[NO]STACK_DEPTH[=n]	/NOSTACK_DEPTH
Design qualifiers:	
/FILL=	"*","^","@",">","o","#","$","x"
/[NO]KEEP	/NOKEEP
/[NO]SCALE[=n]	/NOSCALE
/[NO]TREE	/NOTREE
/[NO]WRAP	/NOWRAP
Tasking qualifiers:	
/[NO]CREATOR_PC	/NOCREATOR
/[NO]PARENT_TASK	/NOPARENT_TASK
/TASK_SWITCH	
CALL-TREE domain qualifiers:	
/[NO]CHAIN_NAME	/NOCHAIN_NAME
/[NO]PATHNAME	/NOPATHNAME
Default qualifiers:	
/DEFAULT	Initial defaults

The PLOT Analyzer command generates performance or coverage histograms. A number of different kinds of data can be plotted along the horizontal axis, and a variety of data domains can be partitioned into buckets along the vertical axis to form the histogram. For example, the node specification PROGRAM_ADDRESS BY ROUTINE selects the program address domain, the domain of all possible program addresses. From this domain, it selects the address ranges of the routines in the program. Each of these address ranges defines a bucket.

When you enter a PLOT command, you must specify the kinds of data to tally, and the way to partition the histogram. These two things define the meanings of the horizontal and vertical axes of the histogram. What kind of data to tally is specified with a datakind qualifier (see the qualifier list).

Each histogram is partitioned into buckets. Each bucket specifies what each individual histogram bar represents. Each bucket is defined by a value range and corresponds to one histogram bar. Each data point in the data file is tallied in the bucket whose value range includes the value of the data point. For a complete description of node specifications and a listing of Analyzer node specifications, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Every PLOT command requires one qualifier from the datakind and the sorting categories to generate a histogram. If you do not specify one qualifier from each category, a default qualifier is used. If your PLOT command has one or more node specifications, the qualifier defaults are taken from the SET PLOT defaults. If your PLOT command does not have a node specification, all qualifier and node specification defaults are taken from the previous PLOT or TABULATE command you entered.

The TABULATE command displays the same information as the PLOT command, but in the form of tables instead of histograms. For a complete description of the PLOT and TABULATE commands, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Examples

1. PCAA> ***PLOT PROGRAM BY MODULE***

   Program counter sampling data is plotted by default. The vertical axis has one histogram bar for each module in the program. It also has one bar for each shareable image used. Each bar indicates how much time is consumed in the corresponding program unit.

2. PCAA> ***PLOT/PAGE_FAULTS/SOURCE MODULE FOO BY LINE***

   The histogram has one bar for each line in module FOO of the user program. The length of each histogram bar is proportional to the number of page faults that occurred at the corresponding line. Because the /SOURCE qualifier is present, the text of each source line is shown next to the histogram bar for that line.

3. PCAA> ***PLOT/NONCOVERAGE/SOURCE MODULE FOO BY CODEPATH***

   The histogram has one bucket for each code path in module FOO. Any code path that is not covered has a full-length histogram bar. Thus, the histogram highlights code that is not tested. The text of each source line is shown next to the corresponding code paths.

4. PCAA> ***PLOT/IO_SERVICES/MAIN_IMAGE PROGRAM BY ROUTINE***

   The number of I/O system service calls is plotted along the horizontal axis. The routines of the program are plotted along the vertical axis. I/O data originating in shareable images (such as the Run-Time Library) charged back to the routines in the main image (the user program) that called the shareable images to cause I/O.

5. PCAA> ***PLOT/SERVICES/DESCENDING/NOZEROS SYSTEM_SERVICES BY SERVICE***

   The names of the OpenVMS system services are plotted along the vertical axis. The number of calls on each system service is plotted along the horizontal axis. The histogram is sorted so the most frequently used services are listed first. All system services that were never called are omitted from the histogram.

6. PCAC> ***PLOT/VCOUNTERS VINSTRUCTION BY VOPCODE***

   This command bases the report view on the disassembled operation code for each vector instruction in the entire application that is sampled. The number of times a vector instruction is used lets you see if your application is spending a lot of time performing certain operations. For example, if you see that the SYNC vector instruction is executed more than any other vector instruction, you can infer that the scalar processor is spending too much idle time waiting for the vector processor to finish an operation.

PRINT

PRINT — Sends the output from the most recent PLOT, TABULATE, INCLUDE, EXCLUDE, LIST,or traverse command to the printer.

Format

PRINT

Command Qualifier	Default
/NOTIFY	None

To print a histogram, table, or listing of raw performance data, enter the command to generate a report on the terminal, and then enter a PRINT command to send that report to the printer. The print file produced by the PRINT command is named PCAPLOT.TMP and is queued to SYS$PRINT.

When printing histograms and tables, the PRINT command assumes that there are 66 lines per page and formats the output accordingly. You can change the number of lines per page by defining the logical name SYS$LP_LINES at DCL level.

To be notified at your terminal when your output has finished printing, use the /NOTIFY qualifier.

SAVE

SAVE — Preserves the contents of an existing screen display in a new screen display.

Format

SAVE old-displayname AS new-displayname[,old-displayname AS new-displayname...]

The SAVE command saves a snapshot of an existing screen display in a new display for later reference. The new display has the same textual contents, attributes, and characteristics as the existing display, except it does not appear on the screen. Using the DISPLAY command, you can later recall the saved display to the terminal screen.

Example

```
PCAA> SAVE PLOT AS SAVED_PLOT
```

This command moves the contents of the screen display named PLOT into the newly created screen display named SAVED_PLOT.

SCROLL

SCROLL — Scrolls a display's screen window in a specified direction over the text of the display, which makes additional display contents visible.

Format

SCROLL [display-name]

/BOTTOM	None
/DOWN[: n]	/DOWN
/LEFT[: n]	None
/RIGHT[: n]	None
/TOP	None
/UP[: n]	None

The SCROLL command is used to view display contents that are not currently visible in a display's screen window. Using the SCROLL command, you can move the screen window up or down over the text of the display. You can also move it left or right. By scrolling, you can see all parts of a large display, even though only a small part might fit in the display's window at any one time.

The easiest way to use the SCROLL command is to use the predefined scrolling keys (see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide) on the numeric keypad.

Examples

1. PCAA> *SCROLL/RIGHT*

   This command scrolls the window to the right by eight columns over the current scrolling display so you can see the rest of long source lines.

2. PCAA> *SCROLL/UP:4 OUT*

   This command scrolls the window up through the OUT display by four lines.

SEARCH

SEARCH — Searches the source code for a specified string, then displays the source lines containing an occurrence of that string in the output window.

Format

SEARCH [range] [string]

/ALL	None
/IDENTIFIER	None
/NEXT	/NEXT
/STRING	/STRING

The SEARCH command enables you to search your source code for a specified string. The source lines containing an occurrence of that string are then displayed. You can use the SEARCH and TYPE commands to display specific lines, or a range of lines. However, if you have moved your source files, you must establish a source directory search list with the SET SOURCE command in order to use these commands. For more information on the SEARCH command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Example

```
PCAC> SEARCH/STRING/ALL COBOLTEST "D"\40:50
40: 02    D2N    COMP-2 VALUE -234560000000
41: 02    D      COMP-2 VALUE  222222.33
47: 02    DRO    COMP-2 VALUE  0.1
50: 02    DR5    COMP-2 VALUE  0.000001
```

This command searches for all occurrences of the letter D in lines 40 through 50 of the module COBOLTEST, and displays the result.

SELECT

SELECT — Selects a screen display as the current output, plot, source, scrolling, error, or input display.

Format

SELECT [display-name]

/ERROR	None
/INPUT	None
/OUTPUT	None
/PLOT	None
/PROMPT	None
/SCROLL	/SCROLL
/SOURCE	None

Attributes are used to select the current scrolling display and direct various types of PCA output to particular displays. This gives you the option of mixing or isolating different types of information, such as PCA input, output, diagnostic messages, and so on, in scrollable displays.

You can use the SELECT command with one or more qualifiers, such as /ERROR or /SOURCE, to assign one or more corresponding attributes to a display. If you do not specify a qualifier, the /SCROLL qualifier is assumed by default.

In general, if you use the SELECT command without specifying a display name, the attribute assignment indicated by the command qualifier is canceled. To reassign display attributes, you must use another SELECT command. See the individual qualifier descriptions in the Parameter and Qualifier Glossary section for details.

Examples

1. PCAC> **SELECT/SOURCE/SCROLL SRC2**

   This command selects display SRC2 as the current source and scrolling display.

2. PCAC> **SELECT/INPUT/ERROR OUT**

   This command selects display OUT as the current input and error display. This causes PCA input, output (assuming OUT is the current output display), and PCA diagnostic messages to be logged in the OUT display in the correct sequence.

3. PCAA> **SELECT/SCROLL OUT**
   PCAA> **SCROLL/UP**

   The first command selects the OUT display as the current scrolling display. The SCROLL command has no *display-name* parameter, and therefore scrolls the OUT display.

4. PCAA> **SELECT/PLOT**

   This command cancels the plot display. All PLOT command output is directed to the output display.

SET

SET — Establishes a variety of definitions, data collection options, and mode settings. These settings are established by the keyword specified on the command line.

Format

SET keyword [parameters]

SET is not an individual command. Each SET command consists of the word SET followed by a required keyword. For details, see the individual SET command descriptions.

SET ANC

SET ANC — Determines which coverage-set points are acceptably noncovered, and saves that information for the next test run.

Format

SET ANC [nodespec]

The SET ANC command lets you select particular coverage-set points as acceptably noncovered, and saves that information for the next test run.

There are two ways to specify ANC information. One is to generate a noncoverage or ANC plot or table, then use a traverse command (such as NEXT), or the FIND command, to pinpoint a particular noncovered point. Line the pointer up with the desired bucket, then enter the SET ANC command at the Analyzer prompt. The SET ANC command saves that address as acceptably noncovered. You can continue traversing the plot in this manner until you have saved all the noncovered points. After you enter the first traverse command, you can enter Ctrl/N to move the pointer from one noncovered point to another, or use the KP COMMA to enter the SET ANC command for you.

Another way to save ANC information is to provide a node specification on a SET ANC command, as in the example provided. For complete information about acceptable noncoverage and an example, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Example

```
PCAA> SET ANC ROUTINE R2 BY LINE, ROUTINE R4 BY LINE
```

This command saves all noncovered lines in routines R2 and R4 as acceptably noncovered.

SET COUNTERS

SET COUNTERS — Establishes the collection of exact execution counts at specified program locations.

Format

SET COUNTERS [nodespec [,nodespec...]]

Command Qualifier	Default
/[NO]STACK_PCS	/NOSTACK_PCS

The SET COUNTERS Collector command establishes execution counters at all program locations specified by the node specifications. After you enter the GO command, the Collector counts the

number of times each specified program location is executed and records each execution in the performance data file. Execution counters are implemented with breakpoint instructions.

To place an execution counter at an individual line, you must use the formal name of the corresponding line node. The format is as follows:

```
LINE [module-name\]%LINE n
```

In this context, *module-name* is the name of the module containing the line and *n* is the line number. You can specify the appropriate routine name instead of the module name as long as the routine name is unique. These path-name qualifications are described in more detail in the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide. The line number is always the compiler-assigned listing line number, which is the same line number the OpenVMS Debugger uses.

If you enter the SET COUNTERS command before entering a SET DATAFILE command, the Collector establishes a default data file. The default data file has the same file name as your program's executable image file and a file type of .PCA.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or / STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Examples

1. PCAC> ***SET COUNTERS/STACK_PCS LINE PRIMES\%LINE 22***

   This command causes PCA to count the executions of line 22 of module PRIMES when data collection starts, and enables the collection of stack PC values. For all further SET COUNTERS requests, a default of /STACK_PCS will be established.

2. PCAC> ***SET COUNTERS PROGRAM BY ROUTINE***

   This command specifies that execution counters be placed at all routine entry points in the user program.

3. PCAC> ***SET COUNTERS/STACK_PCS***

   This command enables the collection of stack PC values for counters.

SET COVERAGE

SET COVERAGE — Establishes the collection of test coverage data at specified program locations.

Format

SET COVERAGE [nodespec [,nodespec...]]

/ANC	None
/PREVIOUS	None

/[NO]STACK_PCS	/NOSTACK_PCS
/UNTIL:n	/UNTIL:1

The SET COVERAGE Collector command establishes the collection of test coverage data for all program locations specified by node specifications. The Collector places a breakpoint at each specified program location to measure test coverage. After you enter the GO command, the Collector determines whether each specified program location is reached during program execution. When a test coverage breakpoint is reached during program execution, the Collector records that program location in the performance data file. The Collector then removes the breakpoint so the overhead of measuring coverage is not incurred in subsequent executions of the same program location.

If you enter the SET COVERAGE command before entering a SET DATAFILE command, the Collector establishes a default data file. The default data file has the same file name as your program's executable image file and a file type of .PCA.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or /STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Examples

1. `PCAC> `*`SET COVERAGE/STACK_PCS PROGRAM BY CODEPATH`*

   This command specifies that test coverage be measured for every code path in the user program, and enables the collection of stack PC values.

2. `PCAC> `*`SET DATAFILE/APPEND COV_TEST.PCA`*
   `PCAC> `*`SET COVERAGE/PREVIOUS PROGRAM BY CODEPATH`*

   The previous SET DATAFILE command specifies that the coverage data from this collection run be appended to an existing performance data file, if such a file exists. The SET COVERAGE command specifies that test coverage be measured for every code path in the user program. The /PREVIOUS qualifier specifies that if an existing data file is found, the Collector will ignore the node specification, if supplied, and will use the breakpoint table from the previous collection run recorded in that data file. Hence, test coverage is measured only once for each code path in the program over all collection runs recorded in the performance data file.

3. `PCAC> `*`SET COVERAGE/UNTIL:4 MODULE NICK BY CODEPATH`*

   This command specifies that test coverage be measured for every code path in module NICK. Up to four executions of each code path are recorded in the performance data file before the coverage breakpoint is removed.

SET CPU_SAMPLING

SET CPU_SAMPLING — Enables the collection of program counter sampling data that occurs during virtual process or CPU time.

Format

SET CPU_SAMPLING

Command Qualifiers	Defaults
/INTERVAL:n	/INTERVAL:10
/[NO]STACK_PCS	/STACK_PCS

The SET CPU_SAMPLING Collector command enables the collection of PC values based on virtual process or CPU time. When you enter the SET CPU_SAMPLING command, PCA will collect PC values only whenever there has been one or more clock ticks on the process clock.

There are many external factors that can affect the behavior of a program in relation to the system (or wall) clock. For example, page faulting and system service wait time (including I/O wait time) make it difficult to determine whether the program counter contains a specific location because of the structure of the program's algorithm, or because of other operations occurring in that interval. Under these conditions, sampling the PC values based on the process CPU time would be more effective and reproducible, because the effects caused by contending processes are reduced.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or /STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

The Collector places the CPU Sampling AST at an elevated mode to enable the sampling of user mode ASTs.

Examples

```
PCAC> SET CPU_SAMPLING/INTERVAL=20/NOSTACK_PCS
```

This command causes the interval to be 20 milliseconds, and disables STACK_PCS collection for this measurement. For all subsequent SET CPU_SAMPLING requests, a default of /NOSTACK_PCS will be assumed.

SET DATAFILE

SET DATAFILE — Identifies the performance data file to be used by the Collector or processed by the Analyzer.

Format

SET DATAFILE [file-spec]

/APPEND	None
/EXECUTABLE	None
/SHAREABLE=(img-name)	None

In the Collector, the SET DATAFILE command identifies the performance data file the Collector should use to record symbol information and performance or coverage data. No performance or coverage data is actually written to the specified file until you enter the GO command. You can specify the data file again with another SET DATAFILE command if you need to correct the data file specification.

If you want to append the data from the present collection run to an existing performance data file, use the /APPEND qualifier. If you want to measure the performance of a shareable image, use the /SHAREABLE qualifier.

In the Analyzer, the SET DATAFILE command identifies a performance data file to be processed by the Analyzer. Data from this file is used by all subsequent PLOT, TABULATE, LIST, and SHOW commands. You can use the SET DATAFILE command to change the performance data file any number of times during the Analyzer session. The Collector builds the performance data file to contain all symbol information and all performance or coverage data required by the Analyzer.

Examples

1. `PCAC> ` **`SET DATAFILE [MY_DIR]TEST`**

    This command specifies that the performance or coverage data be written to a new file called [MY_DIR]TEST.PCA.

2. `PCAA> ` **`SET DATAFILE [MY_DIR]TEST`**

    This command specifies that performance or coverage data should be taken from file [MY_DIR]TEST.PCA by subsequent PLOT, TABULATE, and LIST commands.

3. `PCAC> ` **`SET DATAFILE/APPEND [MY_DIR]OLDTEST`**

    This command specifies that the performance or coverage data be appended to the existing data file [MY_DIR]OLDTEST.PCA.

4. `PCAC> ` **`SET DATAFILE/SHAREABLE=(MYSHR) SHRDATA`**

    This command specifies that the performance of shareable image MYSHR be measured. The name of the data file is SHRDATA.PCA.

SET DISPLAY

SET DISPLAY — Creates new screen displays to be shown on the screen.

Format

SET DISPLAY display-name [AT window-spec] [display-kind]

[, display-name [AT window-spec] [display-kind]...]

/[NO]DYNAMIC	/DYNAMIC
/HIDE	None
/POP	/POP
/PUSH	None
/REMOVE	None
/SIZE:n	None

The SET DISPLAY command creates new screen displays so more displays than the default PLOT, OUT, SRC, and PROMPT displays can appear on your screen. By defining multiple plot displays and selecting each one in turn as the current plot display, you can simultaneously display several different histograms or tables on the screen. Similarly, you can create multiple output displays.

A window in screen mode is a rectangular region on the screen through which you view a display. Its size is defined by a starting line number and by the number of text lines that you want to view through that window.

The screen on a VT-series terminal consists of 24 lines by 80 or 132 columns of text. The screen can also consist of up to 100 lines by 255 columns, depending on the terminal model and the terminal height and width settings.

The pasteboard is similar to a drawing board on the screen to which windows are attached. Each window is pasted onto the pasteboard in the order that it is referenced, and each display is placed in the window with which it is associated. The most recently referenced display is pasted last onto the pasteboard. In addition, as you create more displays, the later displays that you have created might overlay part or all of your earlier displays. What you see on the screen is the final appearance of the pasteboard.

Examples

1. PCAA> *SET WINDOW TOP AT (1,8,45,30)*
   PCAA> *SET DISPLAY NEWINST AT TOP PLOT*
   PCAA> *SELECT/PLOT NEWINST*

   The SET WINDOW command creates a window named TOP starting at line 1, column 45, and extending down for 8 lines and to the right for 30 columns. The SET DISPLAY command creates an instruction display named NEWINST to be displayed through TOP. The SELECT/PLOT command selects NEWINST as the current PLOT display.

2. PCAA> *SET DISPLAY/SIZE:500 PING AT T1*
   PCAA> *SELECT/OUTPUT PING*
   PCAA> *LIST ALL*

   This SET DISPLAY command creates an output display named PING, which is placed at window T1, the top third of the screen. PING holds up to 500 lines of output. The SELECT command selects PING as the current output display. The output of the LIST command is written to PING.

3. PCAA> *SET DISPLAY PANG AT H1 PLOT*
   PCAA> *SELECT/PLOT/SCROLL PANG*
   PCAA> *PLOT MODULE FOO BY LINE*
   PCAA> *SCROLL/DOWN*

   The SET DISPLAY command creates a plot display to be shown at window H1, the top half of the screen. The SELECT command selects that display as the current plot and scrolling displays. The output of the PLOT command is shown in the PANG display and the SCROLL command scrolls that same display. By creating new plot displays, you can see the output from several PLOT or TABULATE commands at the same time, each in its own screen display.

4. PCAA> *SET DISPLAY TINY AT (1,2)*

   This command creates an output display named TINY. TINY is placed at a screen window that starts at line 1 (the top line of the screen) and contains two lines of text.

SET EDITOR

SET EDITOR — Determines the editor used to edit the source file displayed by the current PLOT or TABULATE command.

Format

SET EDITOR [command-line]

/CALLABLE_EDT	None
/CALLABLE_LSEDIT	None
/CALLABLE_TECO	None
/CALLABLE_TPU	None
/[NO]START_POSITION	See text.

The SET EDITOR Analyzer command lets you determine which editor to invoke and how to invoke it when you enter the EDIT command. These editors must be accessible from DCL level. If you do not specify an editor with the SET EDITOR command, the VSI Language Sensitive-Editor is invoked by default. For more information about the SET EDITOR command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Example

```
PCAA> SET EDITOR/CALLABLE_LSEDIT/START_POS "LSEDIT/READ_ONLY"
```

This command causes subsequent EDIT commands to invoke callable LSEDIT with the default command line of LSEDIT/READ_ONLY. The /START_POSITION qualifier will be appended to the command line, which causes the editing session to start on the source line to which the Analyzer is currently pointing.

SET EVENT

SET EVENT — Establishes the collection of event markers for a specified event when specified program locations are executed.

Format

SET EVENT event-name node-spec[,node-spec . . .]

The SET EVENT Collector command establishes the collection of event markers for a specified event when you execute program locations associated with that event. An event marker is a data record written to the performance data file to mark that some significant event has occurred during program execution. You determine which events are significant and the names to assign to those events.

When you enter a SET EVENT command, you must specify the name of the event you want to define. You must also specify the node specifications to associate with that event. The Collector places a breakpoint on each location and, when the breakpoint is reached, writes an event marker for that event to the performance data file.

If you enter the SET EVENT command before entering a SET DATAFILE command, the Collector establishes a default data file. The default data file has the same file name as your program's executable image file and a file type of .PCA.

You can use one event marker per program location. Multiple program locations are enabled for each event name.

Examples

1. `PCAC> `***`SET EVENT COMPUTE LINE OPENIN\%LINE 22`***

    This command establishes the collection of an event marker for event COMPUTE each time line 22 in routine OPENIN is executed.

2. `PCAC> `***`SET EVENT E1 ROUTINE Y, LINE %LINE 42, MODULE X BY ROUTINE`***

    This command establishes the collection of an event marker for event E1 each time line 42 in routine Y is executed, and routine Y is executed in MODULE X.

SET FILTER

SET FILTER — Defines a new filter for data points to be tallied by the PLOT or TABULATE commands.

Format

SET FILTER filter-name filter-spec [,filter-spec...]

Command Qualifiers	Defaults
/[NO]CREATOR_PC	/NOCREATOR_PC
/[NO]CUMULATIVE[=n]	/NOCUMULATIVE
/[NO]MAIN_IMAGE[=prog-unit]	/NOMAIN_IMAGE
/[NO]PARENT_TASK	/NOPARENT_TASK
/[NO]STACK_DEPTH=n	/NOSTACK_DEPTH

The SET FILTER Analyzer command creates a new filter definition. When a subsequent PLOT or TABULATE command is entered, the restrictions specified in the filter definition are applied to all data points in the performance data file. A data point passes a filter if it satisfies any one filter restriction in the filter. Only those data points that pass all defined filters are tallied in the histogram or table.

By using multiple restrictions in a single filter, you OR the restrictions; if a data point passes any one restriction, it passes the filter. By using multiple filters, you AND restrictions; a data point must pass all defined filters to be tallied in a histogram or table.

Filter restrictions that do not apply to a given data point are ignored. A file-name restriction, for example, is ignored for page fault data because no file name is collected for a page fault data point.

To use the /MAIN_IMAGE, /STACK_DEPTH, and /CUMULATIVE qualifiers, you must first gather stack PC values using the Collector SET STACK_PCS command.

If you specify /MAIN_IMAGE=prog-unit, /STACK_DEPTH= n, and /CUMULATIVE= n, they are applied in the following order:/MAIN_IMAGE, /STACK_DEPTH, then /CUMULATIVE. Therefore, if you specify the /MAIN_IMAGE=prog-unit and /STACK_DEPTH= n qualifiers, the Analyzer will search for the first stack PC value within the specified program unit. It will not use the data point in this address. However, the Analyzer uses the data point in the address n call frames on the stack from the main image address when it checks address filter restrictions. If you include /CUMULATIVE on that command, the same action occurs, but the Analyzer also uses the PC values in all the call frames below that (or in the first n call frames down, if /CUMULATIVE= n).

Table 1.5 lists the filter restrictions on the SET FILTER command.

Table 1.5. Filter Restrictions on the SET FILTER Command

Restriction [1]	Explanation
CHAIN_NAME=(chain_name) [/qualifier]	The PC value of a data point must be within the specified comma list of program unit names. This list must be delimited by parentheses. Program units can be expressed as line numbers, routine names, module names, hexadecimal ranges, or shareable images. Wildcards are allowed for whole program units, but not for their identifiers.
FAULT_ADDRESS= n[: m]	The faulting address value of a page fault data point must be n, or in the address range specified by n:m. Both n and m are byte addresses specified in hexadecimal.
FAULT_ADDRESS=prog-unit	The faulting address value of a page-fault data point must be in the address range of the specified program unit. Here, *prog-unit* specifies the name of a module, routine, or line.
FILE_KEY= n[: m]	The file key number of an I/O data point must be in n, or in the address range specified by n:m.
FILE_NAME=file-spec	The file name of an I/O data point must match the name given by *file-spec*. The file specification can include wildcard characters (asterisks). You can specify the device name assigned with SYS$ASSIGN, or MBA n for the mailbox created with SYS$CREMBX, where n represents the number of times SYS$CREMBX was called. The device name and the mailbox must end with a colon.
FILE_VBN= n[: m]	The file virtual block number of an I/O data point must have the value n, or be in the range n:m.
IO_SERVICE=io_service-name	The system service name of an I/O data point must match *io_service-name*.
PROGRAM_ADDRESS=%ANC [/qualifier]	The program counter value of a data point must be found in the coverage set, and must be declared as acceptably noncovered (ANC).
PROGRAM_ADDRESS= n[: m] [/qualifier]	The PC value of a data point must be n, or in the address range specified by n:m. Both n and m are byte addresses specified in hexadecimal.
PROGRAM_ADDRESS=prog-unit [/qualifier]	The PC value of a data point must be in the address range of the specified program unit. Here, *prog-unit* specifies the name of a module, routine, or line. Program address selection qualifiers can be used.
PHYSICAL_IO_COUNT= n[: m]	The physical I/O (read plus write) count for an I/O data point must be n, or in the address range specified by n:m.

Restriction [1]	Explanation
READ_COUNT= *n*[: *m*]	The physical read count for an I/O data point must be *n,* or in the address range specified by *n:m.*
RECORD_SIZE= *n*[: *m*]	The record size in bytes for an I/O data point must be *n,* or in the address range specified by *n:m.*
RUN_NAME= *n*[: *m*]	The data point must be from a collection run *n,* or within the range specified by *n:m.*
RUN_NAME=run-name	The data point must be from a collection run whose name matches *run-name.* The *run-name* specification can include wildcard characters (asterisks).
SERVICE=service_name	The system-service name of a system-services data point must match *service_name.*
TASK_IDENTIFIER= instance_of_task[/ qualifier]	The data point must come from a segment of the data file associated with the task.
TASK_PRIORITY= task_priority	The data point must come from a segment of the data file associated with the task at the given priority level.
TASK_TYPE=task_type [/qualifier]	The data point must come from a segment of the data file associated with the declared type of the task object.
TIME= *n*[: *m*]	The CPU time stamp of a data point must equal *n* milliseconds, or be in the millisecond range specified by *n:m*. CPU time stamps are always relative to the start of the collection run.
TIME=event-name	The data point must come from a segment of the datafile associated with the *event-name* event marker.
VLENGTH= *n*	Filters the VLENGTH domain to show only the Vector Length Register values of a specified length.
VMASK= *n*	Filters the VMASK domain to show the Vector Mask Register values with the specified number of cells enabled.
VOPCODE=vector-opcode	Filters the vector instruction operation code to include only the specified operation codes. You must supply mnemonics of the operation codes to be included in the plot.
VOPERATIONS= *n*	Restricts views based on the number of actual operations performed by vector instructions.
VREGISTER=vector-reg	Filters the vector registers to show only specific registers.
VSTRIDE= *n*	Filters the vector-stride length values to include only the specified values.

Restriction [1]	Explanation
WRITE_COUNT= *n*[: *m*]	The physical write count for an I/O data point must be *n,* or in the address range specified by *n:m*.

[1]Where the equal sign (=) is used to specify the inclusion of data, you can also specify the not equal sign (<>) to specify exclusion of data.

Examples

1. ```
   PCAA> SET FILTER F1 RUN=1
   PCAA> PLOT PROGRAM BY ROUTINE
   ```

   The SET FILTER command defines a filter named F1. When the PLOT command is executed, only data from collection run 1 is tallied in the histogram.

2. ```
   PCAA> SET FILTER FUMBLE PROGRAM_ADDRESS=FRATZ
   PCAA> PLOT/COUNTERS/CUMULATIVE PROGRAM BY ROUTINE
   ```

   The SET FILTER command defines filter FUMBLE. To pass this filter, the program address value of a data point must fall in the address range of routine FRATZ. The PLOT command plots the number of times each routine in the program calls routine FRATZ, directly or indirectly, by way of other routines.

3. ```
   PCAA> SET FILTER F1 RUN=3:5, RUN=8
   PCAA> SET FILTER F2 TIME=COMPUTE
   PCAA> SET FILTER F3 IO_SERVICE=SYS$PUT, IO_SERVICE=SYS$GET
   PCAA> TABULATE/IO_SERVICES MODULE FRATZ BY LINE
   ```

   The SET FILTER commands define filters F1, F2, and F3. To pass filter F1, a data point must come from collection run 3, 4, 5, or 8. To pass filter F2, a data point must be collected after a COMPUTE event marker and before any other event. To pass filter F3, an I/O data point must represent a SYS$PUT or SYS$GET system service call. To be tallied in the table produced by the TABULATE command, an I/O data point must pass all three filters. (Had non-I/O data been tabulated, the data points would need to pass only filters F1 and F2, because the IO_SERVICE restriction is not applicable to non-I/O data.)

4. ```
   PCAA> SET FILTER VLEN1 VLEN=0:7
   PCAA> PLOT PROGRAM BY ROUTINE
   ```

   This example shows locations in the program where the use of vector instructions is not efficient, which indicates that scalar instructions would be more appropriate (that is, for vectors of length 7 or less).

5. ```
   PCAA> SET FILTER VOPS1 VOPERATIONS=0:15
   PCAA> SET FILTER VLEN=32
   PCAA> PLOT PROGRAM BY ROUTINE
   ```

   The SET FILTER command defines the filter VOPS1. This command shows, for vectors of length 32, how many have at least half of their elements being operated on.

6. ```
   PCAA> SET FILTER REGS1 VREGISTER=V1
   PCAA> SET FILTER REGS2 VREGISTER=V2
   PCAA> PLOT INSTRUCTION BY VOPCODE
   ```

   The SET FILTER commands define the filters REGS1 and REGS2. The PLOT command would show which instructions use both registers V1 and V2.

SET IO_SERVICES

SET IO_SERVICES — Enables the collection of I/O services data for each I/O system service call in the program.

Format

SET IO_SERVICES

Command Qualifier	Default
/[NO]STACK_PCS	/STACK_PCS
/NOUSER_ROUTINE	/NOUSER_ROUTINE
/USER_ROUTINE=(image, exit)	/None
/USER_ROUTINE= (image, entry, exit)	None
/USER_ROUTINE= (image, exit)	None

The SET IO_SERVICES Collector command enables the collection of I/O system services data. After you enter the GO command with I/O data collection enabled, the Collector intercepts every I/O system-service call during program execution. For each such call, the Collector gathers the appropriate information from the File Access Block (FAB), Record Access Block (RAB), or other parameters, and records it in the performance data file. The exact information gathered depends on the I/O service, but includes the program address of the call, the name of the service, and the CPU time stamp. It might also include the name of the file accessed, the record size of the transfer, the physical I/O read and write counts, and other information.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or /STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Example

```
PCAC> SET IO_SERVICES/STACK_PCS
```

This command enables the collection of I/O services data and specifies that stack PC values be collected.

SET KEY

SET KEY — Changes the current key definition state. Key definition states are used by keypad keys defined with the DEFINE/KEY command.

Format

SET KEY

/[NO]LOG	/LOG
/[NO]STATE[=state-name]	/NOSTATE

When you define keypad keys using the DEFINE/KEY command, you can assign a specific state name to each key definition. The Collector-defined state names are DEFAULT, GOLD, and BLUE. If

the assigned state is not set when you press the key, the command string to which the key is bound is not processed. You use the SET KEY command to change the current state so keys for the specified state are processed.

If you do not use the SET KEY command, the current keypad state is DEFAULT.

Example

PCAC> *SET KEY/STATE=BLUE*

The SET KEY command sets the current state to BLUE. After entering this command, when you press a keypad key with a BLUE definition, that definition is executed (even if you do not press PF4 first). PF4 is the Collector-defined BLUE key.

SET LANGUAGE

SET LANGUAGE — Establishes the current language setting.

Format

SET LANGUAGE language-name

The language setting determines how PCA parses symbol names in command input. If the language is set to C or C++, PCA treats symbol names as case sensitive. If the language is set to anything other than C or C++, symbol names are assumed to be case insensitive.

The following languages are supported:

	Ada	CXX (C++)	PL/I
	BASIC	DIBOL	RPG
	BLISS	FORTRAN	SCAN
	C	MACRO	UNKNOWN
	COBOL	PASCAL	

When you use the SET DATAFILE command, the language setting is determined by the language of the main routine in the program. Typically, you need not change this setting. However, if you have a mixed-language program that includes C modules, you must change the language setting with the SET LANGUAGE command before you can reference symbols that include lowercase letters.

SET LOG

SET LOG — Specifies that user input and PCA output be recorded in a log file.

Format

SET LOG [file-spec]

The SET LOG command records all PCA input and output in the specified log file. Input is recorded as entered. Output is recorded with a leading exclamation point (!). Because the exclamation point is the PCA comment character, you can rerun the logged PCA session by using the log file as a command procedure.

Example

`PCAC>` ***`SET LOG [PING]PONG`***

This command establishes the file [PING]PONG.LOG as the Collector log file. All subsequent commands and their output are recorded in this file.

SET MAX_SOURCE_FILES

SET MAX_SOURCE_FILES — Specifies the maximum number of source files that PCA can keep open at any one time.

Format

SET MAX_SOURCE_FILES n

Opening a source file requires the use of an I/O channel, which is a limited system resource. Using the SET MAX_SOURCE_FILES command, you can specify how many I/O channels PCA can use to read source files. If PCA reaches this limit, it must close one source file before it can open and read another one. In such cases, source display still works, but at some performance cost.

By default, PCA keeps up to ten source files open at any one time. Typically, there is no need to change this value.

Example

`PCAA>` ***`SET MAX_SOURCE_FILES 20`***

This command sets the number of source files PCA can keep open at one time to 20.

SET MODE

SET MODE — Establishes or cancels various mode settings that affect screen mode.

Format

SET MODE keyword [,keyword]

The SET MODE command sets or cancels screen mode and various submodes within screen mode. Keypad mode is enabled by default. You must disable keypad mode only if your program requires the use of the numeric keypad for its own purposes.

Example

`PCAA>` ***`SET MODE SCREEN, LOG_SCREEN`***

This command enables both screen mode and the logging of screen images in the log file. Screen images are logged only if output logging in general is enabled with the SET LOG command.

SET NOTE

SET NOTE — Stores an informative note about the current collection run in the data file.

Format

SET NOTE "quoted-string"

The SET NOTE Collector command enables you to describe the current collection run in a quoted string following standard PCA rules for quoted strings. This description is stored in the data file. You can retrieve the note by entering the SHOW RUN_DESCRIPTION command in the Analyzer, or the SHOWNOTE command in the Collector.

Example

```
PCAC> SET NOTE "G-FLAT"
```

This command describes the current collection run and stores the quoted string in the data file.

SET PAGE_FAULTS

SET PAGE_FAULTS — Enables the collection of page fault data.

Format

SET PAGE_FAULTS

The SET PAGE_FAULTS Collector command enables the collection of page-fault data. After you enter the GO command with page-fault data collection enabled, the Collector records in the performance data file every page fault that occurs during program execution. The Collector records the program counter value, the faulting virtual address, and a CPU time stamp for each page fault.

Note

You cannot collect stack PC values with page-fault data.

In addition, the Collector can place its own ASTs in elevated mode, thereby supporting the collection of data from user AST routines.

Example

```
PCAC> SET PAGE_FAULTS
```

This command enables the collection of page fault data.

SET PC_SAMPLING

SET PC_SAMPLING — Enables the collection of program counter sampling data.

Format

SET PC_SAMPLING

Command Qualifiers	Default
/INTERVAL: *n*	/INTERVAL:10
/[NO]STACK_PCS	/STACK_PCS

The SET PC_SAMPLING Collector command enables the collection of program counter sampling data. After you enter the GO command with program counter sampling enabled, the Collector samples the program counter (PC) every 10 milliseconds and records each observed program counter value in the performance data file. You can change the sampling interval with the /INTERVAL: *n* qualifier.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or / STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Note

The Collector places the CPU Sampling AST at an elevated mode to enable the sampling of user mode ASTs.

Example

```
PCAC> SET PC_SAMPLING/STACK_PCS
```

This command enables the collection of program counter sampling data, and specifies that stack PC values are collected for this measurement.

SET PLOT

SET PLOT — Establishes default qualifiers and node specifications for the PLOT and TABULATE commands.

Format

SET PLOT [nodespec [,nodespec...]]

See the PLOT command for a full listing of applicable qualifiers and their meanings.

The SET PLOT Analyzer command sets the default qualifiers and node specifications used by the PLOT and TABULATE commands. Qualifiers explicitly specified on the PLOT or TABULATE command override the current defaults.

Example

```
PCAA> SET PLOT/SOURCE/DESCENDING/NOZEROS
```

After this SET PLOT command is entered, all subsequent PLOT and TABULATE commands use the /SOURCE, /DESCENDING, and /NOZEROS qualifiers unless these qualifiers are explicitly overridden.

SET PROMPT

SET PROMPT — Changes the PCA prompt string from PCAC> or PCAA> to a string of your choice.

Format

SET PROMPT string

Example

```
PCAC> SET PROMPT MYNAME> MYNAME>
```

This command changes the Collector prompt from PCAC> to MYNAME>.

SET RUN_NAME

SET RUN_NAME — Establishes the name of the current collection run.

Format

SET RUN_NAME run-name

The SET RUN_NAME Collector command establishes a name for the current collection run. When you enter the GO command, this name is recorded in the performance data file. If you do not enter a SET RUN_NAME command, the null run name (the name with zero characters) is recorded. You can use the collection run name to filter performance or coverage data in the Analyzer.

Examples

```
PCAC> SET RUN_NAME PC_SAMPLE_RUN
```

This command specifies that the collection run name is PC_SAMPLE_RUN. This run name is recorded in the performance data file when you enter the GO command.

SET SEARCH

SET SEARCH — Sets the SEARCH command's default qualifiers. This function establishes current qualifiers and parameters to be used by PCA in the absence of SEARCH command qualifiers.

Format

SET SEARCH [parameter,...]

/ALL	None
/IDENTIFIER	/None
/NEXT	/NEXT
/STRING	/STRING

The SET SEARCH command enables you to set the SEARCH command's default qualifiers. If you provide either two parameters or qualifiers that are mutually exclusive (such as the ALL and NEXT parameters), the latter parameter or qualifier in the sequence takes precedence. If you provide a qualifier and a parameter that are mutually exclusive (such as the /ALL qualifier and the NEXT parameter), the parameter takes precedence. Each of the qualifiers can be expressed as a parameter.

Example

```
PCAC> SET SEARCH IDENTIFIER
PCAC> SHOW SEARCH
      Default search qualifiers:  /NEXT  /Identifier
```

This command sets the default SEARCH qualifier as /IDENTIFIER.

SET SERVICES

SET SERVICES — Enables the collection of system-services data.

Format

SET SERVICES

Command Qualifiers	Defaults
/[NO]STACK_PCS	/STACK_PCS
/NOUSER_ROUTINE	/NOUSER_ROUTINE
/USER_ROUTINE=(image, exit)	/None
/USER_ROUTINE= (image, entry, exit)	None
/USER_ROUTINE= (image, exit)	None

The SET SERVICES Collector command enables the collection of system-services data. When you enter a GO command with system-services data collection enabled, the Collector intercepts every system-service call in the user program and records it in the performance data file. For each call, the Collector records the name of the system service, the program location where it was called, and a CPU time stamp.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or /STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Example

```
PCAC> SET SERVICES/NOSTACK_PCS
```

This command specifies the collection of system-services data and disables the collection of stack PC values.

SET SOURCE

SET SOURCE — Directs PCA to a specified list of directories for source files when source text must be displayed.

Format

SET SOURCE dirname [,dirname...]

/EDIT	None
/MODULE=modname	None

The SET SOURCE command directs PCA to a specified list of directories for source files when source text must be displayed. By default, PCA expects a source file to be in the same directory and in the same exact file it was in when it was compiled. If that file has been moved to another location, you must use the SET SOURCE command to specify that location. You must also establish a directory search list with the SET SOURCE command if you want to view source files in a source window

when in screen mode. For more information on viewing source files in display windows, see the VSI
DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Examples

1. PCAC> *SET SOURCE [PROJA], [PROJB], DISK:[PETER.PAN]*

    This command locates all source files by searching directories [PROJA], [PROJB], and DISK:
    [PETER.PAN], in that order.

2. PCAC> *SET SOURCE/MODULE=NICK [TSAKERES]*

    This command searches directory [TSAKERES] when accessing source files for module NICK.

SET STACK_PCS

SET STACK_PCS — Enables the collection of stack PC values (return addresses from the call stack)
when data is gathered.

Format

SET STACK_PCS

The SET STACK_PCS Collector command enables the collection stack PC values for sampling
data, CPU sampling data, system-services data, I/O services data, exact execution count data, and
test coverage data. The stack PC values for a data point consist of the program counter value and all
call stack return addresses for that data point. In addition, you can use the /STACK_PCS qualifier to
collect stack PC values for these datakinds, or use the /NOSTACK_PCS qualifier when you do not
want to collect stack PC values.

If you collect stack PC values in the Collector, the Analyzer is able to charge back the resources
consumed in shareable images (such as the run-time library) to the code in your program that calls the
shareable images. To use the stack PC information, use the /MAIN_IMAGE, /CUMULATIVE, or /
STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Example

PCAC> *SET STACK_PCS*

This command enables the collection of stack PC values.

SET TASKING

SET TASKING — Instructs the Collector to gather tasking information.

Format

SET TASKING

The SET TASKING Collector command instructs the Collector to gather tasking information. The
Analyzer can then associate the gathered data with the task that caused that data to be gathered. Any
datakind can be plotted against the TASK domain. This enables you to plot the amount of time or
other resource used in various tasks.

To get the Analyzer to report on data by task, specify the SET TASKING command in the Collector, then specify the appropriate domain in the Analyzer by providing that domain's node specification on a PLOT or TABULATE command. For more information on this command and information on creating histograms and tables for Ada-specific domains, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Example

```
PCAC> SET TASKING
```

This command causes tasking data to be collected on all tasks defined in the data file.

SET TERMINAL

SET TERMINAL — Changes the terminal width and page size that PCA uses to format output.

Format

SET TERMINAL

/PAGE:n	None
/WIDTH:n	None

The SET TERMINAL command sets the terminal width or page size that PCA uses to format output. It does not reset the terminal itself; you must do that manually. If you are in screen mode, the screen is refreshed.

The default terminal width and page size are determined by your device type, or by the last SET TERMINAL command you entered at DCL level.

Example

```
PCAA> SET TERMINAL/WIDTH:132
```

This command specifies that the terminal width is 132 columns.

SET TITLE

SET TITLE — Specifies the text of the title lines on each page of PLOT and TABULATE output.

Format

SET TITLE "title-text" [,"title-text"...]

Command Qualifier	Default
/LINE=n	None

The SET TITLE Analyzer command establishes up to six lines of title text to be displayed on each page of output from PLOT and TABULATE commands. The new title does not take effect until you enter another PLOT or TABULATE command (it does not affect the current plot or table). Each title line is centered within the current terminal width. After centering, the rightmost 10 columns of the

first title line are overlaid with the page number. To prevent truncation, the first title line should be limited to the current terminal width, minus 20 characters of text.

If you have established a multiline title, you can change individual lines within that title by using the /LINE= *n* qualifier. If/LINE= *n* is not used, the new title completely replaces the previous title.

If you do not enter a SET TITLE command, "VSI Performance and Coverage Analyzer" is shown on the first title line.

Examples

1. PCAA> *SET TITLE "Welcome to Wonderland", "Enjoy your visit"*

   This command establishes the specified two-line title for the output of subsequent PLOT and TABULATE commands.

2. PCAA> *SET TITLE/LINE=2 'Come Again, Good Day', 'Area "END"'*

   This command replaces line 2 of the title established in the previous example and adds a third line. The resulting three-line title is displayed on the output of subsequent PLOT and TABULATE commands.

SET UNALIGNED

SET UNALIGNED — Enables the collection of unaligned, access-fault data.

Format

SET UNALIGNED

Command Qualifier	Default
/[NO]STACK_PCS	/STACK_PCS

On OpenVMS Alpha systems only, the SET UNALIGNED Collector command enables the collection of unaligned, access-fault data. After you enter the GO command with unaligned, access-fault data collection enabled, the Collector records in the performance data file all unaligned, access-fault data that occurs during program execution. The Collector records the alignment-faulting program counter value, the alignment-faulting virtual address, and other information.

Note

The PCA Collector does not record unaligned, access-fault data if the user program uses image alignment fault reporting.

Note

You must collect stack PC information if you want to use a /MAIN_IMAGE, /CUMULATIVE, or /STACK_DEPTH qualifier with the PLOT or TABULATE command in the Analyzer.

Example

PCAC> *SET UNALIGNED/STACK_PCS*

This command enables the collection of unaligned data and specifies the collection of stack PC values.

SET VCOUNTERS

SET VCOUNTERS — On OpenVMS VAX systems only, the SET VCOUNTERS command establishes the collection of vector processor instruction counts at specified program locations.

Format

SET VCOUNTERS [nodespec [,nodespec...]]

Command Qualifier	Default
/[NO]STACK_PCS	/NOSTACK_PCS

The SET VCOUNTERS Collector command enables the counting of vector processor instructions. The collected data can help you determine to what extent the vector processor is being used. You must specify at least one node specification on the command line to indicate the domain of the data to be collected.

The following node specifications are used when collecting vector instruction counts:

- PROGRAM ADDRESS by VINSTRUCTION

- MODULE module-name by VINSTRUCTION

- ROUTINE routine-name by VINSTRUCTION

Examples

1. PCAC> *SET VCOUNTERS/STACK_PCS PROGRAM_ADDRESS BY VINSTRUCTION*

   This command causes PCA to count the vector instructions based on their addresses when data collection starts, and enables the collection of stack PC values. For all subsequent SET VCOUNTERS requests, a default of /STACK_PCS will be established.

2. PCAC> *SET VCOUNTERS/STACK_PCS PROGRAM_ADDRESS BY VINSTRUCTION*

   This command enables the collection of stack PC values for counters.

SET VCPU_SAMPLING

SET VCPU_SAMPLING — On OpenVMS VAX systems only, the SET VCPU_SAMPLING command enables the collection of vector program counter sampling data that occurs during virtual process or CPU time.

Format

SET VCPU_SAMPLING

/INTERVAL:n	/INTERVAL:10
/[NO]STACK_PCS	/STACK_PCS

The SET VCPU_SAMPLING Collector command enables the collection of PC values for random vector instructions. The collected data lets you determine the scalar and vector parallelism throughout your entire program.

When you collect vector CPU samples, the sampling interval timer includes only the time that the program is actually running the processor. This form of sampling enables you to focus on the particular areas of the program's algorithm where time is being spent, and not on the areas where outside influences consume time.

Example

```
PCAC> SET VCPU_SAMPLING/INTERVAL:20/NOSTACK_PCS
```

This command causes the interval to be 20 milliseconds, and disables STACK_PCS collection for this measurement. For all subsequent SET VCPU_SAMPLING requests, a default of /NOSTACK_PCS will be assumed.

SET VERIFY

SET VERIFY — Causes PCA to verify command input by echoing it to the terminal.

Format

SET VERIFY

The SET VERIFY command causes PCA to echo subsequent command input to the terminal. Input verification is useful when you want to see the commands from command procedures as they are executed. You can disable input verification with the CANCEL VERIFY command.

Example

```
PCAC> SET VERIFY
PCAC> @X.COM
@X.COM
SET DATAFILE [SAGER.TST]PCAPRIME
SET LANGUAGE C
PCAC>
```

The SET VERIFY command enables input verification. The next command entered (@X.COM) is echoed to the terminal. Then a SET DATAFILE and a SET LANGUAGE command are read from the command procedure X.COM. Both commands are echoed to the terminal.

SET VPC_SAMPLING

SET VPC_SAMPLING — On OpenVMS VAX systems only, the SET VPC_SAMPLING command enables the collection of vector program counter sampling data based on the wall clock.

Format

SET VPC_SAMPLING

/INTERVAL:n	/INTERVAL:10
/[NO]STACK_PCS	/STACK_PCS

The SET VPC_SAMPLING Collector command enables the collection of PC values for random vector instructions and causes the sampling rate to be applied to the wall clock. The collected data lets you determine the scalar and vector parallelism throughout your entire program.

When you collect vector PC samples, the sampling interval timer includes all idle time associated with the current run of the program. This form of sampling shows you where the time is being spent in the program, with little cost to the time of actually running the program. You can set the time interval by using the /INTERVAL qualifier.

Example

```
PCAC> SET VPC_SAMPLING/INTERVAL:20/NOSTACK_PCS
```

This command causes the interval to be 20 milliseconds, and disables STACK_PCS collection for this measurement. For all subsequent SET VPC_SAMPLING requests, a default of /NOSTACK_PCS will be assumed.

SET WINDOW

SET WINDOW — Defines screen window names.

Format

SET WINDOW window-name AT

(start-line,line-count,start-column,column-count)

The SET WINDOW command enables you to assign a name to a screen window. A screen window is a region on the screen through which you can view a display. You must specify the window in terms of a starting line number on the screen and a line count. Once defined, you can use a window name in subsequent DISPLAY and SET DISPLAY commands.

PCA provides a substantial number of predefined window names. You can list all the defined window names using the SHOW WINDOW command.

Example

```
PCAA> SET WINDOW TINY AT (1,2)
PCAA> DISPLAY OUT AT TINY
```

The first command defines a window named TINY. The top border line for the window is at line 1 of the screen. The window can display two additional lines of text. The second command displays the OUT screen display through window TINY.

SHOW

SHOW — Displays various definitions, data collection options, and mode settings, depending on the keyword specified on the command.

Format

SHOW keyword [parameters]

SHOW is not an individual command. Each SHOW command consists of SHOW followed by a required keyword. For details, see the individual SHOW command descriptions.

SHOW ALL

SHOW ALL — Displays a summary of all data collection options, and all major Collector and Analyzer mode settings.

Format

SHOW ALL

Command Qualifier	Default
/[NO]FULL	/NOFULL

The SHOW ALL command displays all data collection options, definitions, and mode settings you have established. It also displays the name of the log file, the name of the performance data file, and information on all major mode settings. Entering SHOW ALL is the same as entering a number of more detailed SHOW commands.

Example

```
PCAC> SHOW ALL
Language: FORTRAN
Performance Data File: DEV:[STAR]FFT.PCA;2
Gathering CPU sampling data with STACK_PCS
    AST Timer Interval is 10 ms.
Gathering page fault data
 Default search qualifiers:   /NEXT /STRING
```

This command displays the name of the log file, the current language setting, the name of the performance data file, and all data collection options.

SHOW ANC

SHOW ANC — Displays all the coverage-set points currently set as ANC.

Format

SHOW ANC

SHOW COUNTERS

SHOW COUNTERS — Displays the node specifications for which exact execution counters have been established with previous SET COUNTERS commands.

Format

SHOW COUNTERS

Example

```
PCAC> SHOW COUNTERS
Counters
  Set Counters PROGRAM_ADDRESS BY ROUTINE
  Set Counters LINE PRIMES
\OPENIN\%LINE 22
```

The SHOW COUNTERS command shows that exact execution counts are collected for every routine in the user program and for line 22 of routine OPENIN in module PRIMES.

SHOW COVERAGE

SHOW COVERAGE — Displays the current settings for the SET COVERAGE command.

Format

SHOW COVERAGE

Example

```
PCAC> SHOW COVERAGE
Coverages
   Set Coverage  /Until=1  /Previous  PROGRAM_ADDRESS BY CODEPATH
```

This SHOW COVERAGE command shows that test coverage is measured for every code path in the user program. Each coverage breakpoint is removed the first time it is executed.

SHOW CPU_SAMPLING

SHOW CPU_SAMPLING — Displays whether the collection of program counter sampling data that occurs during the process CPU time is enabled.

Format

SHOW CPU_SAMPLING

Example

```
PCAC> SHOW CPU_SAMPLING
Gathering CPU sampling data with STACK_PCS, AST Timer Interval is 10 ms.
```

The SHOW CPU_SAMPLING command shows that CPU sampling data is collected with stack PCs, the sampling interval is 10 milliseconds, and the AST timer interval is 10 milliseconds.

SHOW DATAFILE

SHOW DATAFILE — Displays the name of the current performance data file.

Format

SHOW DATAFILE

Command Qualifier	Default
/[NO]FULL	/NOFULL

Example

```
PCAC> SHOW DATAFILE/FULL
Performance Data File: DEV$:[PCA.TST]CUMU_MAIN_IO.PCA;1
   Created by: PCA V4.6              Last updated by: PCA V4.6
   Created on:25-MAR-1998 15:51:19.07  Last updated on:25-MAR-1998-
```

```
                    15:51:30.62

 Image file: DEV$:[SAGER.TST]PCAPRIME.EXE;19
  Image name: PCAPRIME                        Image  ID: 01
  Image linked on:  8-MAR-1998 13:57:53.14     by Linker: 04-00

  There is 1 collection run
  The DST is 481 bytes long
  Known shareable images:
     FORRTL          LIBRTL          PCA$COLLECTOR    DBGSSISHR
     PCA$PRVSHR      LBRSHR          SMGSHR
```

The /FULL qualifier gives more complete information about the data file.

SHOW DISPLAY

SHOW DISPLAY — Lists all the screen display definitions. The name, screen window, maximum size,and kind of each defined display are listed. Screen displays are created using the SET DISPLAY command.

Format

SHOW DISPLAY

Example

```
PCAA> SHOW DISPLAY
display SRC at Q123, size = 64, removed, dynamic, kind = SOURCE
display PLOT at Q123, size = 64, dynamic, kind = PLOT
display OUT at E7, size = 100, dynamic, kind = OUTPUT
display PROMPT at E8, size = 64, dynamic, kind = PROGRAM
```

This command displays the definitions of the default screen displays SRC, PLOT, OUT, and PROMPT.

SHOW EDITOR

SHOW EDITOR — Displays the current setting of the editor and its current command line.

Format

SHOW EDITOR

Example

```
PCAA> SHOW EDITOR
The editor is CALLABLE_TPU having the command line:
      "TPU"
```

This command displays the setting of the editor and its command line.

SHOW EVENT

SHOW EVENT — Displays the event names and associated node specifications established with previous SET EVENT commands.

Format

SHOW EVENT

Example

```
PCAC> SHOW EVENT
Events
  Set Event COMPUTE LINE PRIMES
\OPENIN\%LINE 22
```

This command shows that an event marker for event COMPUTE is recorded in the performance data file each time line 22 in routine OPENIN is executed.

SHOW FILTER

SHOW FILTER — Displays all the current filter definitions.

Format

SHOW FILTER

The SHOW FILTER Analyzer command lists the name and filter restrictions for each defined filter. Filters determine which data points from the performance data file are tallied by the PLOT and TABULATE commands. You can create filters with the SET FILTER command and delete them with the CANCEL FILTER command.

Example

```
PCAA> SHOW FILTER
Filter definitions:
  Filter F1:
    RUN_NAME = 1:3
    RUN_NAME = 8
  Filter F2:
    TIME = OUTPUT (event marker)
```

The SHOW FILTER command shows that two filters are defined. Filter F1 has two run-name restrictions and filter F2 has an event marker restriction.

SHOW IO_SERVICES

SHOW IO_SERVICES — Displays whether the collection of I/O services data is enabled.

Format

SHOW IO_SERVICES

Example

```
PCAC> SHOW IO_SERVICES
Gathering I/O service data with STACK_PCS
```

The SHOW IO_SERVICES command shows that I/O services data is collected.

SHOW KEY

SHOW KEY — Displays predefined key definitions, or key definitions created with the DEFINE/ KEY command.

Format

SHOW KEY [key-name]

/ALL	None
/[NO]BRIEF	/BRIEF
/DIRECTORY	None
/[NO]FULL	/NOFULL

The SHOW KEY command displays predefined key definitions, or those keypad definitions established with the DEFINE/KEY command. You can use the /DIRECTORY qualifier to list the names of all states (DEFAULT, GOLD, BLUE, EXPAND, MOVE, and CONTRACT) in which there are key definitions.

Example

```
PCAC> SHOW KEY KP2
DEFAULT keypad definitions:
  KP2 = "SHOW ALL/FULL"
```

This command displays the DEFAULT definition for KP2.

SHOW LANGUAGE

SHOW LANGUAGE — Displays the current language setting, which determines how symbol names are parsed. The language setting is initially determined from the language of the main routine in the user program. You change it with the SET LANGUAGE command.

Format

SHOW LANGUAGE

Example

```
PCAC> SHOW LANGUAGE
Language: FORTRAN
```

This command shows that FORTRAN is the current language.

SHOW LOG

SHOW LOG — Shows whether PCA input and output is being recorded in a log file. If logging is enabled, the name of the log file is displayed. Logging is enabled with the SET LOG command.

Format

SHOW LOG

Example

```
PCAC> SHOW LOG
Output logging on file: USRD$:[USER]PCAC.LOG;1
```

This command shows that output logging is enabled and gives the name of the log file.

SHOW MAX_SOURCE_FILES

SHOW MAX_SOURCE_FILES — Displays the maximum number of source files that PCA can keep open at any one time. You can change this value with the SET MAX_SOURCE_FILES command.

Format

SHOW MAX_SOURCE_FILES

Example

```
PCAA> SHOW MAX_SOURCE_FILES Max_source_files: 10
```

This command shows that PCA can keep up to ten I/O channels open for source files at one time.

SHOW MODE

SHOW MODE — Displays whether keypad mode is enabled. Keypad mode is enabled or disabled with the SET MODE command.

Format

SHOW MODE

Example

```
PCAC> SHOW MODE
Mode: noscreen, keypad, scroll, nomovie, nolog_screen
```

This command shows that keypad mode is enabled.

SHOW NOTE

SHOW NOTE — Displays the currently defined note in the Collector session.

Format

SHOW NOTE

The SHOW NOTE Collector command enables you to retrieve the currently defined note in the Collector session.

Example

```
PCAC> SHOW NOTE
```

This command displays the currently defined note stored in the data file of the Collector session.

SHOW PAGE_FAULTS

SHOW PAGE_FAULTS — Displays whether the collection of page fault data is enabled.

Format

SHOW PAGE_FAULTS

Example

```
PCAC> SHOW PAGE_FAULTS
Gathering page fault data
```

The SHOW PAGE_FAULTS command shows that page fault data is collected.

SHOW PC_SAMPLING

SHOW PC_SAMPLING — Displays whether the collection of program counter sampling data is enabled.

Format

SHOW PC_SAMPLING

Example

```
PCAC> SHOW PC_SAMPLING
```

The SHOW PC_SAMPLING command shows that program counter sampling data is collected with stack PC values, the sampling interval is 10 milliseconds, and the AST timer interval is 10 milliseconds.

SHOW PLOT

SHOW PLOT — Displays the current SET PLOT and active plot default qualifiers and node specifications.

Format

SHOW PLOT

The SHOW PLOT Analyzer command shows both the SET PLOT and the current active plot defaults.

Example

```
PCAA> SHOW PLOT
Plot and tabulate defaults:
  Qualifiers:
    /PC_SAMPLING /DESCENDING /NOMINIMUM /NOMAXIMUM
    /NOCUMULATIVE /SOURCE /ZEROS /NOSCALE /NOCREATOR_PC
```

```
    /NOPATHNAME /NOCHAIN_NAME /WRAP /NOPARENT_TASK /NOKEEP /NOTREE
    /FILL=("*","O","x","@",":","#","/","+")
    /NOSTACK_DEPTH /NOMAIN_IMAGE
  Node specifications:
    PROGRAM_ADDRESS BY ROUTINE
Setting for current plot:
  Qualifiers:
    /PC_SAMPLING /DESCENDING /NOMINIMUM /NOMAXIMUM
    /NOCUMULATIVE /SOURCE /ZEROS /NOSCALE /NOCREATOR_PC
    /NOPATHNAME /NOCHAIN_NAME /WRAP /NOPARENT_TASK /NOKEEP /NOTREE
    /FILL=("*","O","x","@",":","#","/","+")
    /NOSTACK_DEPTH /NOMAIN_IMAGE
  Node specifications:
    TIME BY EVENT
```

The SHOW PLOT command displays the SET PLOT defaults first. The qualifiers and node specifications of the currently active plot are shown next.

SHOW RUN_DESCRIPTION

SHOW RUN_DESCRIPTION — Displays the kinds of data collected for each collection run in the performance data file. This information constitutes a condensed version of what the SHOW ALL command would have shown you in the original Collector sessions.

Format

SHOW RUN_DESCRIPTION parameter

Examples

1. PCAA> *SHOW RUN_DESCRIPTION 2:3*
```
   Collection number 2.  Named - "FOO"
     Events
       Set Event MARK_ROUTINE PRIME\PRIME
     Gathered PC sampling data, AST Timer Interval is 10 ms.
   Collection number 3.  Named - "COVERAGE"
     Coverages
       Set Coverage /Until=1 PROGRAM_ADDRESS BY ROUTINE
```

   This command describes what data was gathered in collection runs 2 through 3. Run 2 gathered program counter sampling data and used an event marker. Run 3 gathered test coverage data.

2. PCAA> *SHOW RUN_DESCRIPTION **
```
   Collection number 1.  Named - ""
     Counters
       Set Counters PROGRAM_ADDRESS BY ROUTINE
     Gathered I/O service data
     Gathering stack PC data
   Collection number 2.  Named - "event_marker"
     Gathered PC sampling data, AST Timer Interval is 10 ms.
   Collection number 3.  Named - "COVERAGE"
     Coverages
       Set Coverage /Until=1 PROGRAM_ADDRESS BY ROUTINE
```

   This command describes all the collection runs. The asterisk is a wildcard character and matches all the run names.

SHOW RUN_NAME

SHOW RUN_NAME — Displays the current collection run name, which is established with the SET RUN_NAME command.

Format

SHOW RUN_NAME

Example

```
PCAC> SHOW RUN_NAME
Current run name = "STEVE"
```

This command shows that the name of the current collection run is STEVE.

SHOW SEARCH

SHOW SEARCH — Displays the current default SEARCH qualifiers.

Format

SHOW SEARCH

Example

```
PCAC> SET SEARCH ALL, STRING
PCAC> SHOW SEARCH
        Default search qualifiers:  /ALL/STRING
```

This command shows the current default SEARCH command qualifiers.

SHOW SELECT

SHOW SELECT — Displays the names of the current output, plot, source, and current scrolling display. These settings are established using the SELECT command.

Format

SHOW SELECT

EXAMPLE

```
PCAA> SHOW SELECT
screen mode not set
display selections:
    scroll = PLOT
    input  = none
    output = OUT
    error = PROMPT
    source = SRC
    program = PROMPT
```

```
    prompt = PROMPT
    plot = PLOT
```

The SHOW SELECT command shows that the screen display OUT is the current output display, the display PLOT is the current scrolling display, and so on.

SHOW SERVICES

SHOW SERVICES — Displays whether the collection of system services data is enabled.

Format

SHOW SERVICES

EXAMPLE

```
PCAC> SHOW SERVICES
Gathering system service data with STACK_PCS
```

The SHOW SERVICES command shows that system services data is collected.

SHOW SOURCE

SHOW SOURCE — Displays the current source directory search lists.

Format

SHOW SOURCE

Command Qualifier	Default
/EDIT	None

The SHOW SOURCE command displays the current, source-directory search lists. The current, source-directory search list determines what directories are searched for source files when source text must be displayed. The source-directory search list is established using the SET SOURCE command.

Example

```
PCAC> SHOW SOURCE
source directory search list for PCAPRIME:
        [MY_DIR]
source directory list for all other modules:
        []
        SRC$
```

This command shows that two source-directory search lists are defined: one for module PCAPRIME and one for all other modules.

SHOW STACK_PCS

SHOW STACK_PCS — Displays whether the collection of stack PC values is enabled.

Format

SHOW STACK_PCS

Example

```
PCAC> SHOW STACK_PCS
Not gathering stack PC data
```

This command shows that stack PC values are not collected.

SHOW SYMBOL

SHOW SYMBOL — Displays information about symbols in the program address domain, or symbol definitions established with the DEFINE/COMMAND command.

Format

SHOW SYMBOL namespec[, namespec...] [IN scope[, scope ...]]

Command Qualifiers	Defaults
/ADDRESS	/ADDRESS is not in effect
/DEFINED	/DEFINED is not in effect
/DIRECT	/DIRECT is not in effect

The SHOW SYMBOL command displays symbols in the program address domain, or symbols you have defined with the DEFINE/COMMAND command.

You can use the /ADDRESS and /DIRECT qualifiers or the IN clause only if /DEFINED is not specified. If you do not specify an IN clause, the whole program will be the scope for symbol selection.

The /DEFINED qualifier cannot be used in combination with any other qualifiers to the SHOW SYMBOL command.

Examples

1. ```
   PCAA> SHOW SYMBOL/DEFINED *
   Defined commands:
     P = "PLOT/NOZERO/DESCENDING"
     BSGB = "SET FILTER FUM PROGRAM_ADDRESS=MUMBLE_FRATZ"
   ```

   This SHOW SYMBOL command displays the definitions of all symbols defined using DEFINE/COMMAND. Only two symbols, P and BSGB, are defined. Their names and definitions are displayed.

2. ```
   PCAA> SHOW SYMBOL/ADDRESS SCALE
    Module SCALE, language PASCAL
     Address range: 00000200 to 00000297
    Routine SCALE\SCALE
     Address range: 00000200 to 0000020B
   ```

   This SHOW SYMBOL command is used to display address ranges of any program unit named SCALE.

SHOW TASKING

SHOW TASKING — Displays the status of the tasking measurement set with the SET TASKING command. For information on collecting tasking data, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Format

SHOW TASKING

SHOW TERMINAL

SHOW TERMINAL — Displays the terminal width and page size used by PCA when formatting output. You can change these settings with the SET TERMINAL command.

Format

SHOW TERMINAL

Example

```
PCAA> SHOW TERMINAL
Terminal page size: 24, Terminal width: 80
```

This command shows that the Analyzer currently assumes a terminal page size of 24 lines and a terminal width of 80 columns when formatting output.

SHOW TITLE

SHOW TITLE — Displays the current title text established with the last SET TITLE command. This title text is output on top of every page of PLOT and TABULATE output.

Format

SHOW TITLE

Example

```
PCAA> SHOW TITLE
Current title text:
  "Welcome to Wonderland"
  "Enjoy your visit"
```

This command displays the text of the current two-line title.

SHOW UNALIGNED

SHOW UNALIGNED — For OpenVMS Alpha systems only, the SHOW UNALIGNED command displays whether the collection of unaligned, access-fault data is enabled.

Format

SHOW UNALIGNED

Example

```
PCAC> SHOW UNALIGNED
Gathering unaligned access data with stack PCs.
```

This example shows that gathering of unaligned, access-fault data is enabled.

SHOW VCOUNTERS

SHOW VCOUNTERS — For OpenVMS VAX systems only, this command displays the node specifications for which vector instruction counters have been established with previous SET VCOUNTERS commands.

Format

SHOW VCOUNTERS

Example

```
PCAC> SHOW VCOUNTERS
  Set Vector Counters MODULE PRIMES BY VINSTRUCTION
```

The SHOW VCOUNTERS command shows that vector instruction counts are collected for MODULE in the user program PRIMES.

SHOW VCPU_SAMPLING

SHOW VCPU_SAMPLING — For OpenVMS VAX systems only, this command displays whether a collection of vector program counter sampling data that occurs during a process CPU time is enabled.

Format

SHOW VCPU_SAMPLING

Example

```
PCAC> SHOW VCPU_SAMPLING
Gathering Vector CPU sampling data with STACK_PCS, AST time interval is 10
 ms.
```

This command shows that vector CPU sampling data is collected with stack PCs, the sampling interval is 10 milliseconds, and the AST time interval is 10 milliseconds.

SHOW VERIFY

SHOW VERIFY — Displays whether command input is being echoed to the terminal. Input verification is enabled with the SET VERIFY command.

Format

SHOW VERIFY

Example

```
PCAC> SHOW VERIFY
Not verifying commands
```

The SHOW VERIFY command shows that command input verification is disabled.

SHOW VPC_SAMPLING

SHOW VPC_SAMPLING — For OpenVMS VAX systems only, this command displays whether a collection of vector program counter sampling data is enabled.

Format

SHOW VPC_SAMPLING

Example

```
PCAC> SHOW VPC_SAMPLING
Gathering Vector PC sampling data with STACK_PCS, AST Timer Interval is 10
 ms.
```

This command shows that vector PC sampling data is collected with stack PCs, the sampling interval is 10 milliseconds, and the AST Timer Interval is 10 milliseconds.

SHOW WINDOW

SHOW WINDOW — Displays the names and definitions of all defined screen windows. Screen windows are either predefined, or defined with the SET WINDOW command.

Format

SHOW WINDOW

Command Qualifier	Default
/ALL	None

SPAWN

SPAWN — Spawns a subprocess so you can temporarily suspend the PCA session to run another program, or execute DCL commands.

Format

SPAWN [dcl-command]

Command Qualifier	Default
/NOWAIT	None

The SPAWN command acts the same as the DCL command SPAWN. You can use it to break out of the PCA session to edit files, read mail, or do any other tasks you would normally do at DCL level, all without ending your PCA session.

Examples

1. PCAC> **SPAWN MAIL**

   You have 1 new message.

   MAIL> **READ/NEW**
   .
   .
   .
   MAIL> **EXIT**
   %PCA-I-RETURNED, control returned to process MY_PROCESS
   PCAC>

   This command spawns a subprocess that runs the MAIL program. When you exit from MAIL, you return to the original PCA session.

2. PCAC> **SPAWN**
   $ **TYPE MUMBLE.DAT**
   .
   .
   .
   $ **LOGOUT**
           Process MY_PROCESS_1 logged out at 29-FEB-1998 14:10:43.55
   %PCA-I-RETURNED, control returned to process MY_PROCESS
   PCAC>

   This command spawns a subprocess. Because you did not specify a DCL command on the SPAWN command, the subprocess prompts for DCL commands. When you log out of the subprocess, you return to the PCA session.

TABULATE

TABULATE — Processes performance or coverage data and displays the processed data in tabular form.

Format

TABULATE [nodespec [,nodespec...]]

See the PLOT command for a full listing of applicable qualifiers and their meanings.

The TABULATE Analyzer command displays the same information as the PLOT command, but in the form of tables instead of histograms. See the PLOT command and the Guide to VSI DECset Performance and Coverage Analyzer for OpenVMS Systems for a full description of the PLOT and TABULATE commands and their qualifiers.

Examples

1. PCAA> **TABULATE PROGRAM BY ROUTINE**

   This command tabulates program counter sampling data by default because no datakind qualifier is specified. The resulting table has one entry for each routine in the program. Each entry shows the number of program counter values collected from the corresponding routine, what percentage this is of the total number of PC values collected, and how wide the 95% confidence interval is.

2. PCAA> **_TABULATE/COUNTERS/SOURCE MODULE FOO BY LINE_**

   This command tabulates exact execution counts for each line in module FOO. The /SOURCE qualifier causes the text of each source line to be shown next to the table entry.

3. PCAA> **_TABULATE/PAGE_FAULTS TIME BY 100 MSECS_**

   This command generates a table showing the number of page faults in each 100-millisecond time interval since the start of the collection run. The intervals represent CPU time. Inspection of this table might reveal page faulting peaks at various times during program execution. Such peaks often occur when a program switches from one phase of execution to another; the new phase typically requires a different set of code and data pages from those in the previous phase.

4. PCAA> **_TABULATE/IO_SERVICES FILE_KEY BY KEY_**

   This command creates a table showing the number of RMS calls that performed I/O using no key, a primary key, a secondary key, or a higher-order key. Keys are used to access indexed, sequential files. Primary-key access is more efficient than secondary-key access. Thus, if more I/O is done on secondary keys than primary keys, I/O might go faster if the key positions are switched.

5. PCAA> **_TABULATE/PC/TREE PROGRAM BY MODULE, PROGRAM BY ROUTINE_**

```
     20.0%    44    Node :=   MODULE1
     10.0%    22    Node :=   . ROUTINEA
      5.0%    11    Node :=   . ROUTINEB
     40.0%    88    Node :=   MODULE2
     30.0%    66    Node :=   . ROUTINEB
      5.0%    11    Node :=   . . ROUTINEC
```

   This command creates a tree structure report. The format of the tree structure is dependent on the specified node specifications: PROGRAM BY MODULE and PROGRAM BY ROUTINE. All buckets representing a routine in the symbolic name space are logically nested under their respective module bucket.

TYPE

TYPE — Specifies which lines of source code to display in the source window.

Format

TYPE [modname[\line-number[:line-number]]][,...]

The TYPE command enables you to view a line or range of lines in your source code. To use the TYPE command, you might have to establish a source directory search list with the SET SOURCE command.

The *modname* without any line numbers displays the first line. If you provide only one line number, the corresponding line of source code is displayed. The default line number is the number for the line following the last line displayed. To display the next line, enter the TYPE command without a line number and press the Return key. By repeating this sequence, you can read through your source program one line at a time. For more information on the TYPE command, see the Guide to VSI DECset Performance and Coverage Analyzer for OpenVMS Systems.

Example

PCAC> **_TYPE COBOLTEST\160,22:24_**

```
module COBOLTEST
   160: START-IT-PARA.
module COBOLTEST
    22: 02      SC2V2   PIC S99V99    COMP VALUE  22.33.
    23: 02      SC2V2N  PIC S99V99    COMP VALUE -22.33.
    24: 02      CPP2    PIC PP99      COMP VALUE  0.0012.
```

This command displays lines 160 and lines 22 through 24 in the module COBOLTEST.

UNDEFINE/COMMAND

UNDEFINE/COMMAND — Deletes command definitions established with the DEFINE/COMMAND command.

Format

UNDEFINE/COMMAND symbol-name [, symbol-name...]

The UNDEFINE/COMMAND command deletes the specified command symbols. The UNDEFINE/COMMAND command and DELETE/COMMAND command perform the same function.

Example

```
PCAC> UNDEFINE/COMMAND FOO, FEE
```

This command cancels the symbols FOO and FEE.

UNDEFINE/KEY

UNDEFINE/KEY — Deletes key definitions established with the DEFINE/KEY command.

Format

UNDEFINE/KEY

/ALL	None
/[NO]LOG	/LOG

The UNDEFINE/KEY command deletes key definitions established with the DEFINE/KEY command. The UNDEFINE/KEY command and DELETE/KEY command perform the same function.

Example

```
PCAC> UNDEFINE/KEY KP7
```

This command cancels the key definitions for KP7.

Chapter 2. Parameter and Qualifier Glossary

This glossary contains two parts:

- Command parameters (see Section 2.1: PCA Command Parameters
- Command qualifiers (see Section 2.2: PCA Command Qualifiers

The two parts complete the information presented in the PCA command dictionary.

2.1. PCA Command Parameters

This glossary lists the PCA command parameters in alphabetical order.

Glossary

"command-string"

This parameter is used as follows:

- With the DEFINE/COMMAND command, specifies the string of PCA commands to associate with the symbol name. The command string must be enclosed in quotation marks and can consist of a unique abbreviation of a PCA command, a whole command, or a list of commands separated by semicolons.

- With the DEFINE/KEY command, specifies the command string to be processed when you press the defined key. Enclose the command string in quotation marks if it contains a space.

"quoted_string"

Describes the current collection run and stores the description in the data file.

"title-text"

Specifies the text of a title line. The text can be up to 80 characters long and must be enclosed in quotation marks. In a list of **title-text** parameters, the first **title-text** parameter specifies the text for the first title line, the second parameter specifies the text for the second title line, and so on. If you use the /LINE= n qualifier, the first **title-text** parameter specifies the text for line n, the second parameter specifies the text for line $n+1$, and so forth. You can specify up to six lines of title text.

ALL
NEXT
IDENTIFIER
STRING

See the qualifier description list.

column-count

Specifies the number of columns in the window. The default is the current screen width.

command-line

The **command-line** parameter is only used with LSEDIT and DECTPU, and is passed to the respective callable editor with each subsequent EDIT command. If you do not specify a **command-**

line,the default editor is LSEDIT or DECTPU, whichever was last used in an editing session. LSEDIT is the initial default editor. The command line consists of the DCL command line used to invoke the editor, except it should begin with LSEDIT or TPU instead of EDIT/ LSEDIT or EDIT/TPU.

datakind   Specifies a kind of performance or coverage data. All records in the performance data file that contain the specified kind of data are listed. The valid **datakind** parameters are as follows:

Keyword	Description
ALL	Lists all data records
BEGIN_END	Lists collection run begin and end records
COUNTERS	Lists execution counter records
COVERAGE	Lists test coverage records
CPU_SAMPLING	Lists CPU sampling records
EVENTS	List event marker records
IO_SERVICES	Lists I/O system service call records
PAGE_FAULTS	Lists page fault records
PC_SAMPLING	Lists program counter sampling records
RUN_DESCRIPTION	Lists collection run description records
SERVICES	Lists system service call records
SHAREABLE_IMAGES	Lists all shareable image name and address range records
TASKING	Lists tasking records
TASK_SWITCH	Lists task switch records
UNALIGNED	Lists unaligned data records
VCOUNTERS	Lists vector instruction execution counter records
VCPU_SAMPLING	Lists vector CPU sampling records
VPC_SAMPLING	Lists vector program counter sampling records

Note

The keywords VCOUNTERS, VCPU_SAMPLING, and VPC_SAMPLING are OpenVMS VAX-specific keywords. The keyword UNALIGNED is an OpenVMS Alpha-specific keyword.

dcl-command   Specifies a DCL command. If you specify the **dcl-command** parameter, a subprocess is created that executes the specified DCL command. Control is returned to the PCA session when the DCL command terminates.

If you omit the **dcl-command** parameter, the SPAWN command creates a subprocess and attaches the terminal to that process. You can then enter DCL commands at the terminal. When you log out of the subprocess or attach back to the parent process, you can continue your PCA session.

	If the DCL command you specify includes a semicolon, you must enclose the command in quotation marks. Otherwise, the semicolon can be mistaken for a PCA command separator. To include a quotation mark inside the string, enter two consecutive quotation marks.
dirname	Specifies a directory to search. If multiple **dirname** parameters are specified, the specified directories are searched for source files, in order. You can specify an OpenVMS logical name for **dirname**.

Note that **dirname** can consist of one, several, or all the fields in a full file specification, though it typically consists only of a directory name. A full file specification has the following format:

```
node::device:[directory]file-name.file-
type;version-number
```

When specifying any of these fields, you must include the punctuation for that field, as shown in this format.

disp-name

With the EXPAND and MOVE commands, specifies a display to be expanded, contracted, or moved. You can specify any of the following:

• A predefined display: SRC, OUT, PROMPT, or PLOT

• A display previously created with the SET DISPLAY command

• A pseudo-display name: %CURDISP, %CURSCROLL, %NEXTDISP, %NEXTOUTPUT, %NEXTPLOT, %NEXTSCROLL, or %NEXTSOURCE

If you do not specify a display, the current scrolling display, as established by the SELECT/SCROLL command, is used.

With the EXTRACT command, specifies a display to be extracted. You can use the wildcard character (*) in a display name.

display-kind

With the DISPLAY command, specifies a new display kind; OUTPUT, SOURCE, and PLOT are the valid keywords. OUTPUT indicates a regular output display for the SHOW and LIST commands. PLOT indicates a display that holds the output of the PLOT or TABULATE command. The SOURCE display holds the output from the TYPE command. If you omit the **display-kind** parameter, the kind of display is not changed.

With the SET DISPLAY command, specifies the kind of display to create; OUTPUT, PLOT, and SOURCE are the valid keywords. OUTPUT indicates a regular output display to hold the output of SHOW, LIST, and SEARCH commands. PLOT indicates a display to contain the output of PLOT and TABULATE commands. SOURCE indicates a display to contain output from a TYPE command. If you omit the **display-kind** parameter, an output display is created.

display-name

This parameter is used as follows:

- With the CANCEL DISPLAY command, specifies a screen display to be canceled.

- With the DISPLAY command, specifies the name of a screen display that you want to view, or whose characteristics you want to modify. This screen display must be predefined (PLOT, SRC, OUT, or PROMPT), or defined with a previous SET DISPLAY command.

- With the SCROLL command, specifies the screen display to scroll. This display must be a predefined display (PLOT, SRC, or OUT), or a display established with the SET DISPLAY command. If you omit this parameter, the current scrolling display, as specified by the SELECT/SCROLL command, is scrolled.

- With the SELECT command, specifies the display to be selected. You can specify any one of the following, with the restrictions noted in the qualifier descriptions:

  - A predefined display (SRC, OUT, PLOT, or PROMPT)

  - A display previously created with the SET DISPLAY command

  - A pseudo-display name: %CURDISP, %CURSCROLL, %NEXTDISP, %NEXTOUTPUT, %NEXTPLOT, %NEXTSCROLL, or %NEXTSOURCE

  If you omit this parameter and do not specify a qualifier, you cancel the current scrolling display (no display has the scrolling attribute). If you omit this parameter but specify a qualifier(/ INPUT, /SOURCE, and so on), you cancel the current display with that attribute (see the qualifier descriptions).

- With the SET DISPLAY command, specifies the name of a new screen display to create. The display name can consist of letters, numerals, underscores, and dollar signs. It cannot begin with a numeral.

event-name   With the CANCEL EVENT command, specifies the name of an event for which event markers will be canceled.

With the SET EVENT command, specifies the name of an event for which event markers should be recorded. The event name must consist of one or more letters, numerals, underscores, and dollar signs. It cannot begin with a numeral.

file-spec   This parameter is used as follows:

- With the @file-spec command, specifies the command procedure to execute or the device from which to read input. The **file-spec** can be any valid OpenVMS file specification. The default file

type is .PCAC when in the Collector and .PCAA when in the Analyzer.

- With the APPEND command, specifies the file to which the output will be appended. The default file specification is taken from the most recent FILE or APPEND command.

- With the EXTRACT command, specifies the file to which the information will be written. You can specify a logical name. If you specify /SCREEN_LAYOUT to save the configuration of the screen, the default specification for the file is PCA.PCAC in the Collector and PCA.PCAA in the Analyzer. Otherwise, the default file specification is PCA.TXT.

- With the FILE command, specifies the file to which the output is written. The **file-spec** parameter can be any valid OpenVMS file specification. The default file type is .PCALIS.

- With the SET DATAFILE command, specifies the performance data file the Collector should use to record symbol information and performance or coverage data, or that the Analyzer should process. The **file-spec** can be any valid OpenVMS file specification. If no file is specified, the current file is used. The default file type is .PCA.

- With the SET LOG command, specifies the log file to create. All subsequent PCA input and output is recorded in the file until logging is disabled. The **file-spec** is any valid OpenVMS file specification. The default file type is .LOG. An omitted **file-spec** defaults to PCAC.LOG in the Collector, or PCAA.LOG in the Analyzer.

filter-name   This parameter is used as follows:

- With the CANCEL FILTER command, specifies an existing filter to cancel.

- With the SET FILTER command, specifies the name of the filter to define. The filter name can consist of letters, numerals, underscores, and dollar signs. It cannot start with a numeral. If a filter of the same name already exists, the Analyzer cancels the old filter before the new one is defined.

filter-spec   Specifies a filter restriction. Each filter restriction specifies some restriction that a data point must satisfy. If a data point satisfies any one restriction for a given filter, it passes the whole filter.

input-file-spec   Any valid OpenVMS file specification that identifies the files from which you want to merge or take acceptably noncovered (ANC) information. Wildcards can be used in the file name and file type.

key-name   This parameter is used as follows:

- With the DEFINE/KEY command, specifies the key you want to define. See Table 1.4 under this command in the Command Dictionary for a list of standard key names.

- With the DELETE/KEY and UNDEFINE/KEY commands, specifies the name of the key whose definition you want to delete.

- With the SHOW KEY command, specifies the name of the key whose definition you want displayed. See Table 1.4 under the DEFINE/KEY command in the Command Dictionary for a list of the valid key names.

keyword

With the PAGE command, displays the first summary page, or a page relative to the current page. The following **keyword** parameters are valid:

- CURRENT—Redisplays the current page

- NEXT—Displays the page after the current page

- PREVIOUS—Displays the page before the current page

- SUMMARY—Displays the first summary page of the plot or table

keyword

With the CAN command, specifies the data collection option, definition, or mode setting to cancel. The valid keywords in the Collector are as follows:

ALL	DISPLAY	SERVICES	VCOUNTERS
COMMAND	EVENT	SOURCE	VCPU_SAMPLING
COUNTERS	IO_SERVICES	STACK_PCS	VERIFY
COVERAGE	LOG	TASKING	VPC_SAMPLING
CPU_SAMPLING	PAGE_FAULTS	UNALIGNED	WINDOW
DATAFILE	PC_SAMPLING		

The valid keywords in the Analyzer are as follows:

ANC	DISPLAY	PLOT	VERIFY
COMMAND	FILTER	SOURCE	WINDOW
DATAFILE	LOG	TITLE	

keyword

With the SET command, specifies the definition, data collection option, or mode setting to establish. In the Collector, the valid keywords are as follows:

COMMAND	KEY	PROMPT	TITLE
COUNTERS	LANGUAGE	RUN_NAME	UNALIGNED
COVERAGE	LOG	SEARCH	VCOUNTERS
CPU_SAMPLING	MAX_SOURCE_FILES	SERVICES	VCPU_SAMPLING

DATAFILE	MODE	SOURCE	VERIFY
DISPLAY	NOTE	STACK_PCS	VPC_SAMPLING
EVENT	PAGE_FAULTS	TASKING	WINDOW
IO_SERVICES	PC_SAMPLING	TERMINAL	

In the Analyzer, the valid keywords are as follows:

ANC	FILTER	MODE	TERMINAL
COMMAND	KEY	PLOT	TITLE
DATAFILE	LANGUAGE	PROMPT	VERIFY
DISPLAY	LOG	SEARCH	WINDOW
EDITOR	MAX_SOURCE_FILES	SOURCE	

keyword
With the SET MODE command, specifies the mode settings to be enabled or disabled. The following table lists the valid **keyword** parameters.

Keyword	Description
KEYPAD	Enables the use of the numeric keypad for command entry. When PCA is in keypad mode, you can associate keys on the numeric keypad with Collector command definitions. Keypad mode is enabled by default.
NOKEYPAD	Disables the use of the numeric keypad for command entry. You must disable keypad mode when the program you are measuring requires the numeric keypad on your terminal to be in numeric mode.
LOG_SCREEN	Causes screen images to be logged in the log file each time the terminal screen is updated. Screen logging occurs only if output logging is enabled with the SET LOG command.
NOLOG_SCREEN	Disables the logging of screen images. NOLOG_SCREEN is the default mode.
MOVIE	Causes the terminal screen to be updated each time a command line is read, even if the line comes from a command file. In movie mode, you can play a command file on the screen.
NOMOVIE	Causes the terminal screen to be updated only after each line of terminal input. NOMOVIE is the default mode.
SCREEN	Enables screen mode so you see PCA output in the form of screen displays viewed through windows on the terminal screen. Screen mode should be used only on DECstations and VT100-or-higher terminals.

Keyword	Description
NOSCREEN	Disables screen mode. Analyzer output is displayed line by line. NOSCREEN is the default mode.
SCROLL	Causes the screen to be updated for every line of output in screen mode. Thus, the output scrolls through the output window. SCROLL is the default mode.
NOSCROLL	Causes the screen to be updated only after a full command has completed. As a result, you might see only the last part of the command's output in its screen window.

keyword   With the SHOW command, specifies the data collection option or the kind of definition or mode setting to display. The valid keywords in the Collector are as follows:

ALL	KEY	RUN_NAME	TERMINAL
COUNTERS	LANGUAGE	SEARCH	UNALIGNED
COVERAGE	LOG	SELECT	VCOUNTERS
CPU_SAMPLING	MAX_SOURCE_FILES	SERVICES	VCPU_SAMPLING
DATAFILE	MODE	SOURCE	VERIFY
DISPLAY	NOTE	STACK_PCS	VPC_SAMPLING
EVENT	PAGE_FAULTS	SYMBOL	WINDOW
IO_SERVICES	PC_SAMPLING	TASKING	

In the Analyzer, valid keywords are as follows:

ALL	KEY	RUN_DESCRIPTION	SYMBOL
ANC	LANGUAGE	RUN_NAME	TERMINAL
DATAFILE	LOG	SEARCH	TITLE
DISPLAY	MAX_SOURCE_FILES	SELECT	VERIFY
EDITOR	MODE	SOURCE	WINDOW
FILTER	PLOT		

label-name   Specifies a label or bucket name associated with a histogram bar or table entry. The FIND command displays the first histogram or table page that contains this label. If the label is a line number, it must be specified as %LINE *n*, where *n* is the line number. Asterisks in the label name are treated as wildcard characters. An asterisk (*) matches zero or more characters in a bucket name. If the **label-name** parameter is omitted, the next occurrence of the last label name specified is displayed.

language-name   Specifies the name of the language to be set. The valid language names are Ada, BASIC, BLISS, C, COBOL, CXX (C++), DIBOL, FORTRAN, MACRO, PASCAL, PL/I, RPG, SCAN, and UNKNOWN.

line-count   Specifies the number of text lines in the window. This number does not include the top header line.

line-number   This parameter is used as follows:

- With the EDIT command, positions the editor at the specified line number. If you specify a **line-number** without a module name, the module is assumed to be the module at which you are currently positioned in the PLOT or TABULATE source display. If you do not specify **line-number**, the editor is positioned at the line at the center of the current page of PLOT or TABULATE output.

- With the TYPE command, defines the line or range of lines to display in the source window. This must be expressed as a decimal integer (the default is 1).

modname   Specifies the name of a module. This parameter indicates that the lines are contained in that module.

module-name   Positions the editor in the specified module. When you specify a module, you must also specify a line number.

n   This parameter is used as follows:

- With the EXCLUDE command, the parameter *n* is an integer representing the datakind to remove from the plot. This integer must be greater than zero. The datakind placed in the plot by the initial PLOT or TABULATE command is the first datakind, the datakind on the following line is considered to be number two, and so on. The default is the last datakind included.

- With the PAGE command, specifies the page number of the page you want displayed. The first page is page 1. If *n* is larger than the number of the last page, the last page is displayed.

- With the SET MAX_SOURCE_FILES command, specifies the maximum number of source files that PCA can keep open at any one time. The value of *n* cannot exceed 20. The default value is 10.

namespec   Specifies the name of the symbol to show. If **namespec** includes asterisks (*), each asterisk is treated as a wildcard character that matches any zero or more characters. All symbols whose names match **namespec** are shown.

new-displayname   Specifies the name of the new screen display that receives the contents of the **old-displayname** display. The new display name can consist of letters, numerals, underscores, and dollar signs. It cannot start with a numeral.

node-spec   Specifies a node specification as a program location. Use care with the PROGRAM BY LINE or PROGRAM BY CODEPATH node specifications because they require more resources.

nodespec   This parameter is used as follows:

- With the CANCEL ANC command, provides the address locations for which to remove the ANC attribute from the coverage-set points in the current data file.

- With the CANCEL COUNTERS command, specifies a node specification to be canceled. This node specification must have been established with a previous SET COUNTERS command. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for a syntax description of Collector node specifications.

- With the CANCEL COVERAGE command, specifies a node specification to be canceled. This node specification must have been established with a previous SET COVERAGE command. For a syntax description of Collector node specifications, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

- With the CANCEL EVENT command, specifies a node specification to be canceled.

- With the CANCEL VCOUNTERS command, specifies a node specification to be canceled. This node specification must have been established with a previous SET VCOUNTERS command.

- With the PLOT command, specifies a program location. Node specifications define how the vertical axis of a histogram is partitioned into buckets.

- With the SET ANC command, specifies the address range that you want to specify as acceptably noncovered. The **nodespec** must specify one or more program address locations. If a node specification is not specified, SET ANC will use the current plot position established by a traverse command or the FIND command.

- With the SET COUNTERS command, specifies a set of program locations that gather execution counts. For a description of the syntax of node specifications, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

- With the SET COVERAGE command, specifies a set of program locations at which to measure test coverage. For a description of the syntax of node specifications, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

- With the SET PLOT command, specifies a program location. Node specifications define how the vertical axis of a histogram or table is partitioned into buckets. For example, the node specification PROGRAM_ADDRESS BY ROUTINE selects the program address domain, the domain of all possible program

addresses. From this domain, it selects the address ranges of the routines in the program. Each of these address ranges defines a bucket. For a full description of node specifications and a summary of Analyzer node specifications, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

- With the SET VCOUNTERS command, specifies a set of restricted program locations that gather vector-processing instruction counts within the specified range.

- With the TABULATE command, specifies a program location. Node specifications specify how to partition the table into buckets. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for a full description of node specifications and a summary of the Analyzer node specifications.

old-displayname   Specifies the name of a screen display whose contents you want to save.

parameters   Depends on the keyword specified.

process-name   Specifies an existing process to which you want to attach the terminal. If the process name includes spaces, enclose the name in quotation marks.

range   Defines the boundaries of the search. If you do not specify a range, the module most recently referenced with a SEARCH or TYPE command is searched from the line following the last line displayed to the end of the module. The Collector can search only one module at a time. These boundaries can be defined with any of the following formats:

- *modname*—Searches the entire specified module

- *modname \line-number*—Searches the specified module from the specified line number to the end of the module

- *modname \line-number:line-number*—Searches the inclusive interval between the two line numbers of the specified module

run-name   This parameter is used as follows:

- With the SET RUN_NAME command, specifies the name of the current collection run. If **run-name** begins with a numeral or contains any characters other than letters, numerals, underscores, or dollar signs, enclose it in quotation marks.

- With the SHOW RUN_DESCRIPTION command, specifies a collection run name. The run description whose run name matches the **run-name** parameter is displayed. If **run-name** includes asterisks (*), each asterisk is treated as a wildcard character that matches zero or more characters. If **run-name** is

not an alphanumeric name, or if it begins with a numeral, it must be enclosed in quotation marks.

run-number   Specifies a collection run number. The run description for any collection run in the specified range is displayed.

scope   Specifies the name of a module, routine, or lexical block, and might include a path-name qualification.

start-column   Specifies the starting (leftmost) column number of the window (the default is column 1).

start-line   Specifies the starting line number of the window. This line displays the top header line for the screen display using the window. The top line of the screen is line number 1.

string   This parameter is used as follows:

- With the SEARCH command, specifies the sequence of characters to search for in the source. If you do not specify a string, the most recently specified search string is used. If the string contains spaces, tabs, commas, or parentheses, it must be delimited by either double or single quotation marks. If the string delimited by quotation marks has a quotation mark within it, the enclosed quotation mark must be expressed as two consecutive quotation marks (" "). If the string delimited by single quotation marks has a single quotation mark within it, the enclosed single quotation mark must be expressed as two consecutive single quotation marks ("). The search is not case sensitive. If the string is not delimited, the default range parameter must not be used and the string cannot contain any blanks, tabs, or semicolons.

- With the SET PROMPT command, specifies the string that is to become the new prompt. If the string contains blanks, semicolons, or lowercase characters, it must be enclosed in single or double quotation marks.

subtopic   Specifies a particular qualifier or parameter on which you want further information, or a keyword that gives you information about a range of qualifiers or parameters.

subtree   This parameter is used as follows:

- With the BACK command, applies BACK to subtrees instead of to the list within the subtree. The BACK SUBTREE command places the pointer at the first leaf within the previous subtree.

- With the CURRENT command, displays the current subtree at its first leaf.

- With the FIRST command, applies FIRST to subtrees instead of to the list within the subtree. The FIRST SUBTREE command places the pointer at the first leaf within the first subtree.

- With the NEXT command, applies NEXT to subtrees instead of to the list within the subtree. The NEXT SUBTREE command places the pointer at the first leaf within the next subtree.

symbol-name   This parameter is used as follows:

- With the DEFINE/COMMAND command, specifies a symbol name to associate with a list of PCA commands. The symbol name can be up to 255 characters long and can include letters, numerals, underscores, and dollar signs. It cannot start with a numeral. If **symbol-name** is already defined, the new definition replaces the old definition.

- With the DELETE/COMMAND command, specifies the name of a command symbol whose definition is to be deleted. This symbol name must have been defined by a previous DEFINE/ COMMAND command.

- With the UNDEFINE/COMMAND command, specifies the name of a command symbol whose definition is to be deleted.

topic   Specifies the name of the command or topic with which you need help.

window-name   This parameter is used as follows:

- With the CANCEL WINDOW command, specifies the name of a screen window definition to be canceled.

- With the SET WINDOW command, specifies the name of the window to define. The window name can consist of letters, numerals, underscores, and dollar signs. It cannot begin with a numeral.

window-spec   This parameter is used as follows:

- With the DISPLAY command, specifies a new screen window for the display. If you specify **window-spec**, the display is shown in that window. If you omit **window-spec**, the display remains in its current window.

  The **window-spec** can be the name of a predefined window, the name of a window you have defined with a SET WINDOW command, or a window specification of the following form:

  ```
  start-line,line-count[,start-column,column-
  count]
  ```

  The **start-line** is the screen line number at which you want the top border of the display to be placed. The **line-count** is the number of lines of text you want to see in the window. The **start-column** and **column-count** specify the leftmost column and the number of columns in the window, respectively. The **Start-column** and **column-count** default to column 1 and the current screen width, respectively.

- With the SET DISPLAY command, specifies the screen window at which to position the new display. If you omit **window-spec**, the display is positioned at window H1 (top half of screen) or H2 (bottom half of screen).

  The **window-spec** can be the name of a predefined window, the name of a window you have defined with a SET WINDOW command, or a window specification of the following form:

  ```
  start-line,line-count[,start-column,column-
  count]
  ```

  The **start-line** is the screen line number at which you want the top border of the display to be placed. The **line-count** is the number of lines of text you want to see in the window. The **start-column** and **column-count** specify the leftmost column and the number of columns in the window, respectively. The **start-column** and **column-count** default to column 1 and the current screen width, respectively.

2.2. PCA Command Qualifiers

This glossary lists the PCA command qualifiers in alphabetical order.

Glossary

/ADDRESS   Displays the address specification for each selected symbol, when applicable.

/ALL   This qualifier does the following:

- With the CANCEL ANC command, removes the ANC attribute from all coverage-set points in the current data file. You cannot specify a node specification with the /ALL qualifier.

- With the CANCEL COUNTERS command, cancels all execution counters established with earlier SET COUNTERS commands. If you specify the /ALL qualifier, omit the node specifications.

- With the CANCEL COVERAGE command, cancels all test coverage breakpoints established with earlier SET COVERAGE commands. If you specify /ALL, omit the node specifications.

- With the CANCEL DISPLAY command, cancels all screen displays (except PROMPT) including the PCA-defined PLOT, SOURCE, and OUT displays. After canceling all displays, you must use the SET DISPLAY command to create new screens if you want to select them for use in screen mode (see the SELECT command). If you use the /ALL qualifier, omit the **display-name** parameter.

- With the CANCEL EVENT command, cancels the collection of all event markers established with earlier SET EVENT

commands. If you specify the /ALL qualifier, omit the **event-name** and **nodespec** parameters.

- With the CANCEL FILTER command, cancels all currently defined filters. If you use the /ALL qualifier, omit the **filter-name** parameters.

- With the CANCEL VCOUNTERS command, cancels all vector instruction counters established with an earlier SET VCOUNTERS command. If you specify the /ALL qualifier, omit the node specifications. (This is used on OpenVMS VAX systems only.)

- With the CANCEL WINDOW command, cancels all screen window definitions created with the SET WINDOW command, including all window definitions defined by the Analyzer. If you use the /ALL qualifier, omit the **window-name** parameters.

- With the DELETE/KEY and UNDEFINE/KEY commands, deletes all key definitions in the specified state. If you use the /ALL qualifier, omit the **key-name** parameter. If you do not specify a state, all key definitions in the current state are deleted. Use the /STATE qualifier to specify one or more states.

- With the EXTRACT command, extracts all screen displays. If /ALL is used, do not specify a display name. Do not specify SCREEN_LAYOUT with /ALL.

- With the SEARCH command, searches the specified range for all occurrences of the string and displays every line containing the string.

- With the SET SEARCH command, makes /ALL the default SEARCH qualifier.

- With the SHOW KEY command, displays all key definitions in the current state. You can use the /STATE qualifier to request key definitions in other states. If you use the /ALL qualifier, do not specify a key name.

- With the SHOW WINDOW command, displays all PCA- and user-defined windows.

/ALPHABETICALLY[=n[:m]]	Sorts the histogram alphabetically by label names. If you specify $=n$, only the first n buckets (after sorting) are retained in the histogram. If you specify $=n:m$, only buckets n through m (after sorting) are retained.
/ANC	This qualifier does the following:

- With the INCLUDE command, adds the acceptable noncoverage (ANC) datakind to the current buckets.

- With the MERGE command, writes all ANC information to the coverage-set table in the output file. /ANC and /

SPM_PROCESS_ID cannot be used together in the same command. You can use the /ANC qualifier only when merging PCA performance data files. For more information on acceptable noncoverage, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

- With the PLOT command, places the ANC label in the percentage column of every ANC bucket.

- With the SET COVERAGE command, saves code path information in the performance data file for every module containing coverage-set points. This information is used by the Analyzer MERGE/ANC command to validate the ANC information being merged (see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide).

/APPEND   This qualifier does the following:

- With the EXTRACT command, appends the information at the end of the file, rather than creating a new file. By default, a new file is created. Do not specify /SCREEN_LAYOUT with / APPEND.

- With the FILE command, appends the new output to the existing text file specified by the **file-spec** parameter. If no such file exists or if /APPEND was not used, the Analyzer writes the output to a new file.

- With the SET DATAFILE command, appends the performance or coverage data from the current collection run to an existing performance data file. If the file specified by **file-spec** exists, the Collector opens that file and appends the performance or coverage data from the current collection run to it. If the file specified by **file-spec** does not exist, the Collector creates a new data file with that name and writes the performance or coverage data to that file. If you do not specify the /APPEND qualifier, the Collector always creates a new file. You can append data to an existing data file only by using the same image that originally created the file.

/ASCENDING[=n[:m]]   Sorts the histogram in ascending order by bucket data count (smallest count first). If you specify =*n*, only the first *n* buckets (after sorting) are retained in the histogram. If you specify =*n:m*, only buckets *n* through *m* (after sorting) are retained.

/BOTTOM   Scrolls the window down to the bottom of the display's text.

/BRIEF   The /BRIEF qualifier requests that only the key definition and state
/NOBRIEF   be displayed. /NOBRIEF requests that all qualifiers associated with a key definition be displayed. The /BRIEF qualifier performs the same function as the /NOFULL qualifier. /BRIEF is the default.

/CALLABLE_EDT /CALLABLE_LSEDIT /CALLABLE_TECO /CALLABLE_TPU	These qualifiers specify the callable editor invoked by subsequent EDIT commands. If you do not specify a qualifier, the spawned command line is used to invoke the editor.
/CHAIN_NAME /NOCHAIN_NAME	Specifies to use the comma-list form of presentation on the plot for call trees. The /NOCHAIN_NAME default qualifier specifies to use the indentation form. With the /NOCHAIN_NAME qualifier, the PLOT command's maximum and minimum sorting qualifiers are not in effect. If you specify /CHAIN_NAME, /NOZEROS is in effect.
/CLEAR	Erases the entire textual contents of the specified screen displays.
/CODEPATHS	Lists the names of all modules for which code-path data was gathered.

/COUNTERS   This qualifier does the following:

- With the INCLUDE command, adds the counters datakind to the current buckets.

- With the PLOT command, tallies exact execution-count data along the horizontal axis. To use this qualifier, you must have collected execution counts.

/COVERAGE   Tallies test coverage data along the horizontal axis. Every program unit that is executed at least once gets a full-length histogram bar. To use this qualifier, you must have collected test coverage or execution-count data.

/COVERAGE /NONCOVERAGE	Specifies whether coverage or noncoverage datakinds are added to the current buckets.
/COVERAGE_SET	Lists the contents of the coverage-set table.
/CPT	Lists the contents of the code-path table.

/CPU_SAMPLING   This qualifier does the following:

- With the INCLUDE command, adds the CPU_SAMPLING datakind to the current buckets.

- With the PLOT command, tallies CPU sampling data along the horizontal axis. To use this qualifier, you must have collected CPU sampling data.

/CREATOR_PC
/NOCREATOR_PC   This qualifier does the following:

- With the INCLUDE command, specifies whether the data point is charged to the program address that created the task (/CREATOR_PC), or to the address of the data record (/NOCREATOR_PC).

- With the PLOT command, charges a data point to the location in the program that created the task, rather than to the PC value

of the measurement. You can use the /CREATOR_PC qualifier only when the PROGRAM_ADDRESS domain is in effect, and when tasking data has been collected. It overrides the /MAIN_IMAGE, /STACK_DEPTH, and /CUMULATIVE qualifiers. The default is /NOCREATOR_PC.

- With the SET FILTER command, determines whether to use a PC value associated with the creation of the task. /NOCREATOR_PC is the default and specifies not to use the PC value associated with the creation of a task. /CREATOR_PC overrides the /MAIN_IMAGE, /STACK_DEPTH, and /CUMULATIVE qualifiers, and can be used only with the filter address filter specification.

/CUMULATIVE[=n] /NOCUMULATIVE	This qualifier does the following:

- With the INCLUDE command, specifies whether the cumulative tabulation for this datakind only is performed.

- With the PLOT command, tallies all stack PC values for each data point in the histogram. Thus, each bucket shows how much time or other resource is spent in the corresponding program unit and all program units it calls. The /CUMULATIVE qualifier has no effect if you are not plotting against the program address domain. To use the /CUMULATIVE qualifier, you must have collected stack PC values in the Collector. The /NOCUMULATIVE qualifier, which is the default, tallies only the original program counter value for each data point.

- With the SET FILTER command, determines whether a data point passes a filter restriction if the first stack PC value passes the restriction, or if any one stack PC value for that data point passes the restriction. The default, /NOCUMULATIVE, lets a data point pass an address filter restriction only if the first stack PC value passes the restriction. The /CUMULATIVE qualifier requires you to specify the level of call frames from the start that the data point must pass.

/DDIF	Causes the current plot to be stored in DDIF format. This qualifier cannot be used with the /APPEND qualifier when you are using the DECwindows interface.

/DEFAULT	This qualifier does the following:

- With the INCLUDE command, applies the default characteristics from the SET PLOT command to the specified datakind. Only the applicable qualifiers that are following in the list can be applied.

- With the PLOT command, specifies that all default qualifiers and node specifications be taken from the SET PLOT defaults instead of from the currently active plot.

/DEFINED	Displays symbol definitions established with the DEFINE/ COMMAND command. This qualifier cannot be used with other qualifiers.

/DESCENDING[=n[:m]]	Sorts the histogram in descending order by bucket data count (largest count first). If you specify =n, only the first n buckets (after sorting) are retained in the histogram. If you specify =n:m, only buckets n through m (after sorting) are retained. /DESCENDING is the default sorting qualifier.
/DIRECT	Displays symbols that are defined directly in the symbol selection scope. Symbols defined in lexical entities nested within the scope specified by the scope parameters are not shown.
/DIRECTORY	Displays the names of all states for which keys have been defined. The predefined key states are DEFAULT, GOLD, BLUE, EXPAND, MOVE, and CONTRACT.
/DMT	Lists the contents of the Debug Module Table (DMT).
/DOWN[: n]	This qualifier does the following:

- With the EXPAND command, moves the bottom border of the display down by n lines (if n is positive) or up by n lines (if n is negative). If n is omitted, the border is moved down by one line.

- With the MOVE command, moves the display down by n lines (if n is positive) or up by n lines (if n is negative). If n is omitted, the display is moved down by one line.

- With the SCROLL command, scrolls the window down the specified number of lines. If :n is omitted, the window scrolls approximately three quarters of the window height.

/DST	Lists the contents of the Debug Symbol Table (DST).
/DYNAMIC /NODYNAMIC	With the DISPLAY and SET DISPLAY commands, controls whether a display automatically adjusts its window dimensions proportionally when a SET TERMINAL command is entered. The /DYNAMIC qualifier is the default.
/ECHO /NOECHO	Determines whether the equivalence string is displayed on your terminal after you press the defined key. The default is /ECHO. You cannot use the /NOECHO qualifier with the /NOTERMINATE qualifier.
/EDIT	This qualifier does the following:

- With the CANCEL SOURCE command, cancels the EDIT command directory-search list. Use this qualifier only if the EDIT command has its own specific directory-search list.

- With the SET SOURCE command, specifies that the indicated directory-search list is to be used in locating source files for the Analyzer EDIT command.

- With the SHOW SOURCE command, shows the directory-search list established by the last SET SOURCE/EDIT command. This directory-search list is used in locating source files for the EDIT command.

/ERROR   Selects a specified display as the current error display. This causes all PCA diagnostic messages to go to that display. The display specified must be either an output display or the PROMPT display.

If you do not specify a display, the PROMPT display is selected as the current error display. By default, the PROMPT display has the error attribute.

/EVENT   This qualifier does the following:

- With the INCLUDE command, adds the event datakind to the current bucket.

- With the PLOT command, tallies event execution data along the horizontal axis. To use this qualifier, you must have collected events execution data.

/EVENT_NAMES   Lists all the event marker names recorded in the data file.

/EXECUTABLE   Causes the Collector to use the default performance data-file name. The default data-file name consists of the name of the user program's executable image and the .PCA file type. You get the default file name if you do not use the SET DATAFILE command. However, you need the /EXECUTABLE qualifier if you want the default file name and you also want to specify the /APPEND or /SHAREABLE qualifier. If you specify the /EXECUTABLE qualifier, omit the **file-spec** parameter.

/EXIT   The /EXIT qualifier terminates the Analyzer session before invoking
/NOEXIT   the editor. The editor is then run in the same process and you are returned to DCL level when exiting the editor.

The /NOEXIT qualifier returns you to the Analyzer session when you exit the editor. /NOEXIT is the default.

/FAULT_ADDRESS   This qualifier does the following:

- With the INCLUDE command, adds the fault address datakind to the current buckets.

- With the PLOT command, tallies page-fault data along the horizontal axis using the faulting address value to tally each data point. To use this qualifier, you must have collected page-fault data.

/FILE_NAMES   Lists the names of all files opened or created by the user program.

/FILL=("*","O","x","@",":","#","/","+")   Specifies the fill to be used for the first, second, and so on datakind in a plot. You can define a maximum of eight character strings and each must be delimited by double or single quotation marks. The /FILL qualifier is a default design qualifier.

/FILL=   Defines the fill to be used within a PLOT command. You can define a maximum of eight character strings and each must be delimited by double or single quotation marks. The default fill characters are as follows:

```
/FILL="*","^","@",">","o","#","$","x".)
```

/FULL	Gives more complete status information than the unqualified SHOW ALL command. If you specify the /FULL qualifier, you see the status of all data collection options, even those not established, and all mode settings. You also see full data-file information. If you omit the /FULL qualifier, you see only the data-collection options that are established and only a subset of the remaining status information.
/FULL /NOFULL	This qualifier does the following: • With the SHOW DATAFILE command, determines whether more detailed information from the performance data-file header is given. • With the SHOW KEY command, the /FULL qualifier requests that all qualifiers associated with a key definition be displayed. The /NOFULL qualifier requests that only the key definition and state be displayed. The /FULL qualifier performs the same function as the /NOBRIEF qualifier. /NOFULL is the default.
/GST	Lists the contents of the Global Symbol Table (GST).
/HEADER	Lists the performance data-file header.
/HEXADECIMAL	Lists all performance data records in hexadecimal form, in addition to the normal text form.
/HIDE	With the DISPLAY and SET DISPLAY commands, conceals the display under any other displays that overlap it. Any displays that were previously hidden under the specified display become visible.
/IDENTIFIER	This qualifier does the following: • With the SEARCH command, names the string as an identifier. The specified range is searched for the first occurrence of this identifier. If the string (identifier) is bound on either side by a character that can be part of an identifier defined by the SET LANGUAGE command, it is not displayed. The search continues until it finds an occurrence of the string that is not bound by such characters. • With the SET SEARCH command, makes /IDENTIFIER the default SEARCH qualifier.
/IF_STATE=(state-name,...) /NOIF_STATE	Specifies one or more states, one of which must be in effect for the key definition to be processed. If you omit the /IF_STATE qualifier or you use /NOIF_STATE, the current state is used. The **state-name** is an alphanumeric string. States are established with the /SET_STATE qualifier or SET KEY command.
/IMAGE	Lists information about the user program's executable image file.
/INPUT	Selects a specified display as the current input display. This causes that display to echo PCA input (which always appears in the PROMPT display). The display specified must be an output display.

If you do not specify a display, the current input display is canceled and PCA input is not echoed to any display (PCA input appears only in the PROMPT display). By default, no display has the input attribute.

/INTERVAL:n

With the SET CPU_SAMPLING and SET PC_SAMPLING commands, specifies the sampling interval in milliseconds. The value of n must be at least 10 milliseconds and at most 60,000 milliseconds. Because the granularity of the OpenVMS VAX system clock is 10 milliseconds, you should use a multiple of 10. The default sampling interval is 10 milliseconds.

/INTERVAL:n

With the SET VPC_SAMPLING and SET VPCU_SAMPLING commands, specifies the sampling interval in milliseconds. The value of n must be at least 10 milliseconds and at most 60,000 milliseconds. Because the granularity of the OpenVMS VAX system clock is 10 milliseconds, use a multiple of 10. The default sampling interval is 10 milliseconds; therefore, you can use the /INTERVAL: n qualifier only to specify a longer sampling interval.

/IO_SERVICES

This qualifier does the following:

- With the INCLUDE command, adds the I/O services datakind to the current buckets.

- With the PLOT command, tallies the number of I/O system service calls along the horizontal axis. To use this qualifier, you must have collected I/O services data.

/KEEP
/NOKEEP

Instructs the Analyzer to create the plot using all the current datakinds. The /KEEP qualifier causes all secondary datakinds from the previous plot to be kept for the new plot. The /NOKEEP qualifier causes only the primary datakinds to be printed; the secondary datakinds are not retained. The /NOKEEP qualifier is the default.

/LEFT[: n]

This qualifier does the following:

- With the EXPAND command, moves the left border of the display to the left by n lines (if n is positive) or to the right by n lines (if n is negative). If n is omitted, the border is moved to the left by one line.

- With the MOVE command, moves the display to the left by n lines (if n is positive) or right by n lines (if n is negative). If n is omitted, the display is moved to the left by one line.

- With the SCROLL command, scrolls the window left the specified number of columns. If :n is not specified, the window scrolls left by eight columns. You cannot scroll past the first column.

/LINE=n

Specifies that the first **title-text** parameter should go in line n of the title, that the second **title-text** parameter should go in line n+1 of the

title, and so on. This qualifier is used to selectively replace certain lines of a previously established title.

/LOCK_STATE /NOLOCK_STATE	Specifies that the state set by the /SET_STATE qualifier remains in effect until explicitly changed. If you use the /NOLOCK_STATE qualifier, the state set by /SET_STATE is in effect only for the next definable key you press, or for the next read-terminating character you type. The default is /NOLOCK_STATE unless you specify the / TERMINATE qualifier.
/LOG /NOLOG	This qualifier does the following: • With the DEFINE/KEY command, controls whether the system displays a log message indicating that a key definition has been successfully created. The default is /LOG. • With the SET KEY command, controls whether a log message is displayed indicating that the key state has been set. The default is /LOG. • With the UNDEFINE/KEY and DELETE/KEY commands, controls whether a log message is displayed indicating that the specified key definition has been deleted. The default is /LOG.
/LOST_DATA /LOST_DATA=all /LOST_DATA=PCA /LOST_DATA=USER /NOLOST_DATA	Reports lost data in the headers and summary page for the LIST command. The default is ALL data when the /LOST_DATA qualifier is specified. If you specify PCA, it selects only data records lost by PCA. If you specify USER, it selects only data records lost by USER report. /NOLOST_DATA does not report lost data.
/MAIN_IMAGE[=n:m] /NOMAIN_IMAGE	Specifies when checking address filter restrictions whether to use the first stack PC value in the given address range that falls in the main image. By default, the first stack PC value that falls in the main image is used.
/MAIN_IMAGE[=n[:m]] /MAIN_IMAGE[=prog-unit] /NOMAIN_IMAGE	Specifies to use the first call stack PC value that falls in your program's main image when tallying program addresses. You can specify the PC value to be n or the range $n{:}m$. If you specify a program unit or address range, it selects the first PC value in that address. The /NOMAIN_IMAGE qualifier, which is the default, specifies to use the original program counter value to tally each data point.
/MAIN_IMAGE[=prog-unit] /NOMAIN_IMAGE	Performs the main image tabulation for this datakind only. / NOMAIN_IMAGE does not perform the tabulation.
/MAXIMUM=[n] /NOMAXIMUM	Determines whether to omit buckets that exceed a maximum percentage. The /MAXIMUM= n qualifier specifies that buckets whose percentage exceeds $n\%$ should be omitted from the histogram. The /NOMAXIMUM qualifier, which is the default, specifies that no maximum threshold applies.

/MINIMUM=[n] /NOMINIMUM	Determines whether to omit buckets that fall below a minimum percentage. The /MINIMUM= *n* qualifier specifies that buckets whose percentage falls below *n%* should be omitted from the histogram. The /NOMINIMUM qualifier, which is the default, specifies that no minimum threshold applies.
/MODULE=modname	Specifies that the indicated directory search list is to be used in locating source files for the specified module. Other modules do not use this search list.
/MODULE=module-name	Cancels the directory search list for a specific module. Use this qualifier only if the module name has its own specific directory search list.
/MODULE_NAMES	Lists the names of all the modules in the image for which the data was collected.
/NEXT	This qualifier does the following: • With the SEARCH command, searches the specified range for the first occurrence of the string and displays that line. This is the default. • With the SET SEARCH command, makes /NEXT the default SEARCH qualifier.
/NOCOLLECT	Executes the user program without collecting performance or coverage data.
/NOEXPAND	Prevents the traverse command BACK, CURRENT, FIRST, or NEXT from expanding to source.
/NONCOVERAGE	Tallies test noncoverage data along the horizontal axis. Every program unit that is not covered (is not executed at least once) gets a full-length histogram bar. Program units that are covered do not get histogram bars. To use this qualifier, you must have collected coverage or execution-count data.
/NOSORT[=n[:m]]	Specifies that the histogram not be sorted. If you specify =*n*, only the first *n* buckets are retained in the histogram. If you specify =*n:m*, only buckets *n* through *m* are retained.
/NOTIFY	Sends a message to your terminal when the printer has finished your print job.
/NOWAIT	Specifies that PCA should not wait for the subprocess to finish. You can then enter additional PCA commands while the subprocess is running. When the subprocess eventually completes, a message is displayed on your terminal. In general, you should not use the /NOWAIT qualifier if the spawned subprocess prompts for input or displays much output on the terminal.
/OUTPUT	Selects a specified display as the current output display. This causes PCA output that is not already directed to another display to go to that display. The display specified must be either an output display or the PROMPT display.

If you do not specify a display, the PROMPT display is selected as the current output display. By default, the OUT display has the output attribute.

/PAGE:n
Specifies that the number of lines per page should be set to *n* lines. Typically *n* is 24, but you can use any value from 11 to 100.

/PAGE_FAULTS
This qualifier does the following:

- With the INCLUDE command, adds the page fault datakind to the current buckets.

- With the PLOT command, tallies page-fault data along the horizontal axis using the program address value to tally each data point. To use this qualifier, you must have collected page-fault data.

/PARENT_TASK
/NOPARENT_TASK
This qualifier does the following:

- With the INCLUDE command, specifies whether the data point is charged to the task that created the task (/PARENT_TASK), or to the task (/NOPARENT_TASK).

- With the PLOT command, charges a data point to the parent of the current task, rather than to the current task. You can only use the /PARENT_TASK qualifier when the TASK or the TASK_TYPE domain is in effect, and when tasking data has been collected. The default is /NOPARENT_TASK.

- With the SET FILTER command, determines whether to use a PC value associated with the parent task's characteristics. /NOPARENT_TASK is the default and specifies not to use the PCA value associated with the parent task's characteristics. You can use /PARENT_TASK only with the TASK and TASK_TYPE filter specifications.

/PATHNAME
/NOPATHNAME
Specifies to use full path names for each routine name in the plot. The default /NOPATHNAME qualifier specifies to use unqualified routine names.

/PC_SAMPLING
This qualifier does the following:

- With the INCLUDE command, adds the PC sampling datakind to the current buckets.

- With the PLOT command, tallies program-counter sampling data along the horizontal axis. To use this qualifier, you must have collected program-counter sampling data. The /PC_SAMPLING qualifier is the default datakind qualifier.

/PHYSICAL_IO_COUNT
This qualifier does the following:

- With the INCLUDE command, adds the physical I/O datakind to the current buckets.

- With the PLOT command, tallies the total number of physical I/O operations (reads plus writes) along the horizontal axis. To use this qualifier, you must have collected I/O services data.

/PLOT	Directs all output from PLOT, TABULATE, and related commands to the specified display. This display becomes the current plot display. By default, the PLOT display has the plot attribute.
/POP	Places a specified display at the top of the display pasteboard ahead of any other displays. This is the default.
/POP /NOPOP	/POP places a specified display at the top of the display pasteboard ahead of any other displays. This is the default. /NOPOP preserves the order of all displays on the pasteboard.
/POSITION=n	Includes the newly gathered datakind in the specified location of a multiple datakind plot. The parameter *n* must be an integer greater than 1. The integer represents a position in the graph; it directly corresponds to *n* in the EXCLUDE command.
/PREVIOUS	This qualifier does the following:

- With the CANCEL COVERAGE command, cancels the use of uncovered addresses from previous runs. If you specify the /PREVIOUS qualifier, omit the node specification.

- With the SET COVERAGE command, uses the breakpoint table left by the previous collection run when appending test coverage data to an existing performance data file. If such a breakpoint table exists, the node specifications are ignored. In addition, this qualifier causes the Collector to write a table of test coverage breakpoint locations not yet covered (or not yet covered *n* times if /UNTIL: *n* is specified) to the data file at the end of the current collection run.

  This qualifier causes test coverage to be measured only once (or only *n* times for /UNTIL: *n*) for an entire set of collection runs. If a program location has been covered in one collection run, no attempt to measure coverage of that location is made in any subsequent collection run. Using the /PREVIOUS qualifier makes data collection more efficient, but you lose the ability to determine what individual collection runs covered which program locations. If you use the /PREVIOUS qualifier, use the /APPEND qualifier on the SET DATAFILE command.

/PROMPT	Selects a specified display as the current prompt display. This is where PCA prompts for input. Currently, only the PROMPT display can be specified. Note that you cannot cancel the PROMPT display (the PROMPT display always has the prompt attribute).
/PUSH	Has the same effect as /HIDE.
/PUSH	/PUSH has the same effect as /HIDE. /NOPUSH preserves the order of all displays on the pasteboard (it has the same effect as /NOPOP).

/NOPUSH

/READ_COUNT	This qualifier does the following:

- With the INCLUDE command, adds the read count datakind to the current buckets.

- With the PLOT command, tallies the number of physical read operations along the horizontal axis. To use this qualifier, you must have collected I/O services data.

/REFRESH	Refreshes the terminal screen. Do not use parameters or other qualifiers on a DISPLAY/REFRESH command.
/REMOVE	This qualifier does the following:

- With the DISPLAY command, creates a display that does not appear on the screen. Its definition and textual contents are saved, and can be brought back with another DISPLAY command.

- With the SET DISPLAY command, creates a display that does not appear on the screen. Although a removed display is not visible on the screen, its definition still exists, and you can request that it be displayed by using the DISPLAY command.

/REMOVE	Creates a display that does not appear on the screen. Its definition and textual contents are saved, and can be brought back with another DISPLAY command.
/RIGHT[:n]	This qualifier does the following:

- With the EXPAND command, moves the right border of the display to the right by n lines (if n is positive) or to the left by n lines (if n is negative). If n is omitted, the border is moved to the right by one line.

- With the MOVE command, moves the display to the right by n lines (if n is positive) or left by n lines (if n is negative). If n is omitted, the display is moved to the right by one line.

- With the SCROLL command, scrolls the window right the specified number of columns. If :n is not specified, the window scrolls right by eight columns.

/SCALE=[n] /NOSCALE	Determines whether a fixed or adjusted histogram scale is used. The /SCALE= n qualifier allows you to select a fixed scale for the plot. The /NOSCALE default qualifier selects adjusted scaling, which causes the longest bar on the histogram to occupy the full width of the plot.
/SCREEN_LAYOUT	Writes a file that contains the commands describing the current state of the screen. This information includes the screen height and width, and the position, display kind, and display attributes of every existing display. This file can then be executed with the @file-spec command to reconstruct the screen at a later time. Do not specify /SCREEN_LAYOUT with the /ALL or /APPEND qualifiers.

/SCROLL	Selects a specified display as the current scrolling display. This is the default display for the SCROLL, MOVE, and EXPAND commands. Although any display can have the scroll attribute, you can use only the MOVE and EXPAND commands (not the SCROLL command) with the PROMPT display.
	If you do not specify a display, the current scrolling display is canceled and no display has the scroll attribute.
/SEQUENCE_NUMBERS / NOSEQUENCE_NUMBERS	Reports sequence numbers in the headers and summary page for the LIST command. /NOSEQUENCE_NUMBERS does not report sequence numbers.
/SERVICES	This qualifier does the following:
	• With the INCLUDE command, adds the services datakind to the current buckets.
	• With the PLOT command, tallies the number of system service calls along the horizontal axis. To use this qualifier, you must have collected system services data.
/SET_STATE=state-name /NOSET_STATE	Causes the specified state name to be set when the key is pressed. The state name can be any alphanumeric string.
	If you omit the /SET_STATE qualifier or if you use / NOSET_STATE, the current state that was locked remains in effect after you press the defined key. If you have not included this qualifier with a key definition, you can use the SET KEY command to change the current state.
/SHAREABLE=(img-name)	Specifies that you want to measure the performance of a shareable image and that you have created a Debug Symbol Table that fits that shareable image. The **img-name** parameter specifies the shareable image to measure. If you use the /SHAREABLE qualifier, specifying the data file is optional. For more information on how to measure the performance of shareable images, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.
/SHAREABLE_IMAGE	Lists the names of all shareable images used by the user program or the Collector.
/SINCE="time-spec" /BEFORE="time-spec"	Sets the lower and higher limits on the collection times of the SPM data being merged. These qualifiers can only be specified together with the /SPM_PROCESS_ID qualifier. The **time-spec** parameters must be enclosed within double quotation marks ("") and must be valid OpenVMS absolute date/time specifications.
/SIZE:n	This qualifier does the following:
	• With the DISPLAY command, changes the maximum size of a normal output display to *n* lines. If more than *n* lines are written to the display, the oldest lines are lost as the new lines are added. If you omit this qualifier, the maximum size is not changed.

	• With the SET DISPLAY command, sets the maximum size of a normal output display to be *n* lines. If more than *n* lines are written to the display, the oldest lines are lost as the new lines are added. If you omit this qualifier, the maximum size is set to 50 lines. For plot displays, *n* specifies the number of lines of PLOT or TABULATE output to be buffered internally by the Analyzer at one time. There is usually no need to change this number.
/SOURCE	Selects a specified display as the current source display. This causes the output of all TYPE commands to go to that display. The display specified must be a source display. If you do not specify a display, the current source display is canceled and no display has the source attribute.
/SOURCE /NOSOURCE	Determines whether source text is shown next to the histogram bars. The /SOURCE qualifier, which is the default, displays source code in histograms generated with BY LINE or BY CODEPATH node specifications. The /NOSOURCE qualifier specifies that source text not be displayed.
/ SPM_PROCESS_ID=process-id	Merges the specified SPM data files. Each **process-id** parameter must be a valid OpenVMS process identification number specified as a hexadecimal number. You can omit the leading zeros. Each collection of PC data is assigned a RUN_NAME identifier of the form SPM_*n*, where *n* is the OpenVMS process identification number that you specify in the **process-id** list. The data must be collected for the image of the currently open data file. A list of more than one number must be enclosed in parentheses. If you specify the /SPM_PROCESS_ID qualifier, the local node, current default devices, and current default directory are assumed as the default for each input file specifications provided, and default file types are not applied. If you do not specify the /SPM_PROCESS_ID qualifier, the same defaults are assumed, but the file types default to .PCA. The /ANC and /SPM_PROCESS_ID qualifiers cannot be used together in the same command.
/STACK_DEPTH=n /NOSTACK_DEPTH	This qualifier does the following: • With the INCLUDE command, specifies whether the stack is measured to a depth of *n* for this datakind. • With the PLOT command, determines whether the Analyzer will climb the call stack to determine the PC values. To use this qualifier, you must have collected stack PC values. The /NOSTACK_DEPTH qualifier is the default. • With the SET FILTER command, specifies whether to perform stack-depth analysis. /NOSTACK_DEPTH is the default. If you

use the /STACK_DEPTH=*n* qualifier, specify the depth of the call frame that falls below the first stack PC value.

/STACK_PCS
/NOSTACK_PCS

This qualifier does the following:

- With the SET COUNTERS command, specifies whether to collect stack PC values for counter data. The /NOSTACK_PCS qualifier is the default upon entry into the Collector. If you specify /STACK_PCS, subsequent SET COUNTERS requests assume a default of /STACK_PCS. With either qualifier, specifying the node specification is optional.

- With the SET COVERAGE command, specifies whether to collect stack PC values for coverage data. The /NOSTACK_PCS qualifier is the default upon entry into the Collector. If you specify /STACK_PCS, subsequent SET COVERAGE requests assume a default of /STACK_PCS. With either qualifier, specifying the node specification is optional.

- With the SET CPU_SAMPLING command, specifies whether to collect stack PC values for CPU data. The /STACK_PCS qualifier is the default upon entry into the Collector. If you specify /NOSTACK_PCS, subsequent SET CPU_SAMPLING requests assume a default of /NOSTACK_PCS.

- With the SET IO_SERVICES command, specifies whether stack PC values are collected for I/O services data. /STACK_PCS is the default upon entry into the Collector.

- With the SET PC_SAMPLING command, specifies whether stack PC values are collected for PC sampling data. /STACK_PCS is the default upon entry into the Collector.

- With the SET SERVICES command, specifies whether or not to collect stack PC values when gathering system services data. Stack PC values are collected by default.

- With the SET UNALIGNED command, specifies whether stack PC values are collected for unaligned fault data. /STACK_PCS is the default upon entry into the Collector.

- With the SET VCOUNTERS command, specifies whether to collect stack PC values for counter data. The /NOSTACK_PCS qualifier is the default upon entry into the Collector. If you specify /STACK_PCS, subsequent SET VCOUNTERS requests assume a default of /STACK_PCS. With either qualifier, specifying the node specification is optional.

- With the SET VCPU_SAMPLING command, specifies whether to collect stack PC values for vector CPU data. The /STACK_PCS qualifier is the default upon entry into the Collector. If you specify /NOSTACK_PCS, subsequent SET VCPU_SAMPLING requests assume a default of /NOSTACK_PCS.

- With the SET VPC_SAMPLING command, specifies whether to collect stack PC values for vector PC data. The /STACK_PCS qualifier is the default upon entry into the Collector. If you specify /NOSTACK_PCS, subsequent SET VPC_SAMPLING requests assume a default of /NOSTACK_PCS.

/START_POSITION /NOSTART_POSITION	Specifies whether to pass the number of the source line to the editor. The /START_POSITION is appended to the command line that is also passed to the editor with each subsequent EDIT command. /NOSTART_POSITION is the default, unless spawned LSEDIT is selected. In that case, /START_POSITION is the default.
/START_TIME /NOSTART_TIME	Displays system service CPU start times in the LIST output. The /NOSTART_TIME does not display CPU start times.
/STATE=(state-name [,...]) /NOSTATE	This qualifier does the following:

- With the DELETE/KEY and UNDEFINE/KEY commands, specifies the names of the states in which the specified key definitions are to be deleted. If you specify only one state name, omit the parentheses. A state name can be any appropriate alphanumeric string.

  If you omit the /STATE qualifier, or if you use /NOSTATE, key definitions in the current state are deleted.

- With the SHOW KEY command, specifies the name of a state for which the specified key definitions are to be displayed. If you specify only one state name, omit the parentheses. The predefined state names are DEFAULT, GOLD, BLUE, EXPAND, MOVE, and CONTRACT. Additional state names are created with the DEFINE/KEY command.

  If you omit the /STATE qualifier or use /NOSTATE, key definitions in the current state are displayed.

/STATE=state-name/ NOSTATE	Specifies the keypad state to be set. The state name can be any alphanumeric string. If you omit the /STATE qualifier or use /NOSTATE, the current state is left unchanged. The default state is DEFAULT.
/STRING	This qualifier does the following:

- With the SEARCH command, searches the specified range for an occurrence of the string without interpreting the context surrounding the string. This is the default.

- With the SET SEARCH command, makes /STRING the default SEARCH qualifier.

/TASK_NAME	Lists the task name section.
/TASK_SWITCH	This qualifier does the following:

- With the INCLUDE command, adds the Ada datakind to the current buckets.

- With the PLOT command, represents the number of times there was a task context switch. It can be applied to the following domains: TIME BY *n* MSECs, TASK, TASK_PRIORITY, and TASK_TYPE. This qualifier will work only with the TIME, TASK, TASK_PRIORITY, and TASK_TYPE filter specifications.

/TASK_TYPE   Lists the task type section.

/TERMINATE   Enables the current equivalence string processed as command input
/NOTERMINATE   when you press the defined key. The default is /NOTERMINATE, which allows you to enter additional input before the equivalence string is processed. Pressing the Return key after pressing the defined key has the same effect as using /TERMINATE in the key definition.

/TOP   Scrolls the window up to the top of the display's text.

/TREE   Determines whether a tree structure report is created. /NOTREE
/NOTREE   is the default. The format of the tree structure is dependent on the current node specification used.

           For a call tree's node specification, the tree represents the relationship of the dynamic calls. All other node specification's tree structures represent the static relationship of each bucket in the symbolic tree. For example, all buckets representing a routine in the symbolic name space are logically nested under the respective module bucket.

/UNTIL:n   Causes each test coverage breakpoint to be removed the nth time it is reached during program execution. By using the /UNTIL: *n* qualifier, you can determine whether a program location is executed at least *n* times. If you omit the /UNTIL: *n* qualifier, each breakpoint is removed the first time it is reached.

/USER_ROUTINE   Enables users to record system-service arguments or control the
/NOUSER_ROUTINE   recording of system service records by PCA for the SET SERVICES and SET IO_SERVICES commands.

           The /USER_ROUTINE qualifier enables PCA to call a user-written procedure before and after system service calls. The qualifier records system-service arguments and controls PCA's recording of system-service records.

           /USER_ROUTINE parameters include the following:

- Image—The shareable image name.

- Entry—PCA calls this routine before calling the system service. Omit this name if you do not want a routine called.

- Exit—PCA calls this routine after calling system service. Omit this name if you do not want a routine called.

The syntax of the entry routine and exit routine is as follows:

```
FLAG = entry-routine or exit-routine (AP, sequence-
number,
    CPU-time, service-name, service-index, Ptr)
```

The format includes the following items:

- **FLAG**—Indicates whether PCA should write the system service record. PCA writes the record only if you set the low bit. PCA writes a user lost data entry for each nonwritten record.

- **AP**—Indicates the contents of the argument pointer (AP) at the service call by value.

- **sequence-number**—Indicates the PCA sequence number that corresponds to the system service record by value.

- **CPU-time**—Indicates the CPU time of a longword value.

- **service-name**—Indicates the system service name of the address of a counted ASCII string.

- **service-index**—Indicates the system service index.

- **Ptr** indicates a user-supplied longword value.

Note

When you write a routine to use with the /USER_ROUTINE qualifier, note the following rules:

- Write a shareable image, making the entry and exit routine names UNIVERSAL.

- Only one of the entry and exit routines can be null.

- If your routine instructs PCA to not write some data records, the Analyzer signals missing data indications.

/UP[: *n*]	This qualifier does the following:
	• With the EXPAND command, moves the top border of the display up by *n* lines (if *n* is positive) or down by *n* lines (if *n* is negative). If *n* is omitted, the border is moved up by one line.
	• With the MOVE command, moves the display up by *n* lines (if *n* is positive) or down by *n* lines (if *n* is negative). If *n* is omitted, the display is moved up by one line.
	• With the SCROLL command, scrolls the window up the specified number of lines. If *:n* is omitted, the window scrolls approximately three quarters of the window height.
/VCOUNTERS	This qualifier does the following:

- With the INCLUDE command, adds the VCOUNTERS datakind to the current buckets.

- With the PLOT command, tallies exact execution count data of vector instructions along the horizontal axis. To use this qualifier, you must have collected vector execution counts.

/VCPU_SAMPLING   This qualifier does the following:

- With the INCLUDE command, adds the VCPU_SAMPLING datakind to the current buckets.

- With the PLOT command, tallies CPU sampling data along the horizontal axis. To use this qualifier, you must have collected vector CPU sampling data.

/VPC_SAMPLING   This qualifier does the following:

- With the INCLUDE command, adds the VPC_SAMPLING datakind to the current buckets.

- With the PLOT command, tallies program counter sampling data along the horizontal axis. To use this qualifier, you must have collected vector program counter sampling data.

/WIDTH:n   Specifies that the terminal width should be set to *n* columns or spaces. Typically *n* is 80 or 132, but you may use any value from 40 to 255.

/WRAP
/NOWRAP   Specifies to wrap PLOT output that is too long. The /NOWRAP default qualifier specifies to truncate PLOT output that is too long.

/WRITE_COUNT   This qualifier does the following:

- With the INCLUDE command, adds the write count datakind to the current buckets.

- With the PLOT command, tallies the number of physical write operations along the horizontal axis. To use this qualifier, you must have collected I/O services data.

/WRITE_COUNT   Adds the write count datakind to the current buckets.

/ZEROS
/NOZEROS   Determines whether zero data count buckets are omitted from the histogram. The /NOZEROS qualifier specifies that buckets with zero data counts be omitted from the histogram. Thus, the histogram becomes smaller because buckets without data are dropped. The /ZEROS qualifier, which is the default, retains buckets with zero data counts.)

Chapter 3. Error Messages

3.1. Error Messages and Recovery Procedures

This section contains descriptions of the PCA error messages and provides remedial information for recovering from errors.

Message: ABKEYW, ambiguous qualifier or keyword '*text*' – supply more characters

Type: Warning

Explanation: PCA cannot complete the command because '*text*' is ambiguous in the context of this command.

Message: ABTBADST, PCA may be in an inconsistent internal state

Type: Warning

Explanation: The user has typed three or more Ctrl/Cs. The Analyzer normally responds to a Ctrl/C by continuing execution until a consistent internal state is achieved and then aborts the current command. However, if PCA is in an infinite loop and does not detect this consistent state, it will abort anyway after the third Ctrl/C.

User action: Leave and enter PCA again to make sure the internal state is consistent.

Message: ABTCNTLC, command aborted by user request

Type: Warning

Explanation: The current PCA command has been aborted due to a request entered by the user. PCA will attempt to recover gracefully.

User action: Enter another PCA command. It has recovered gracefully to a consistent internal state.

Message: APPFILE, appending to file '*file-name*'

Type: Informational

Explanation: APPEND and FILE commands issue this message to indicate the actual file being appended. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

User action: Verify that the file name in the message is consistent with your desired file.

Message: ATTACHED, terminal now attached to process '*process-name*'

Type: Informational

Explanation: The SPAWN command issues this message before attaching to the new process.

Message: BADCHAR, illegal character encountered

Type: Warning

Explanation: An unanticipated character was found on the command line. This character might be found to have meaning in other syntaxes; however, it lacks meaning in this context.

User action: Verify the appropriate syntax of the specific command.

Message: BADFILNAM, illegal file specification '*text*'

Type: Warning

Explanation: The provided file name does not follow standard OpenVMS file specification conventions.

Message: BADMODE, Bad access mode for secure usage of user routines '*access-mode*'

Type: Warning

Explanation: Before activating or calling a user-routine image, the Collector detected a BAD *access-mode* for secure usage of the user routines and avoided making the activation, or call, for security considerations. The Collector recovers by not activating or calling the user- routine image.

Message: BADOPCODE, cannot decode opcode. Bad table or no instruction starting at PC '*program-location*' in module '*module-name*'

Type: Informational

Explanation: During code-path analysis, the Collector has detected an invalid operation code. This might be because the user's instruction stream contains data that cannot be decoded by the Collector. The code-path information for the module is discarded. The Collector will continue to run. Note that this should only occur within modules assembled with MACRO.

User action: Using the PC, module name, and link map, you can find where PCA cannot decode the operation code. You might then have to work back from there to find where you need to fix your code because many random series of bytes can look like a valid instruction stream.

Message: BADSTACK, stack corrupted - no further data available '*program-location*'

Type: Informational

Explanation: During stack PC gathering, the Collector has detected that it cannot read the next call frame and it has not found the bottom of the stack. The Collector recovers by flagging the stack frame as incomplete in the data file. The Analyzer will then report this as lost data.

Message: BADSTARTPC, cannot access start PC = '*program-location*'. Please submit an SPR.

Type: Error

Explanation: The Collector treats *program-location* as the transfer address of the user program. The Collector cannot use this address to start the execution.

User action: Submit a Software Performance Report (SPR) with the version number of the Collector and a copy of the image. Include any relevant data pertaining to your current environment.

Message: BADSTATUS, bad status returned from '*routine-name*'

Type: Error

Explanation: PCA got an unexpected error status from a system service or RTL routine *routine-name* that it called.

User action: Examine the error message. If the problem is related to a lack of quota or otherwise related to your program's behavior, take corrective action. If you believe that the problem lies in PCA, submit a Software Performance Report (SPR).

Message: BEGINCOL, data collection begins

Type: Informational

Explanation: At the start of the collection, this message is issued before the Collector transfers control to the application program.

User action: Prepare to respond to all of the application's input. To suppress this message, define the logical PCA$INHIBIT_MSG.

Message: BUFOVFLLOST, some data was lost at collection time due to buffer overflow

Type: Informational

Explanation: The data used for this plot or tabulation might be incomplete as a result of a data buffer overflow condition at collection time.

User action: See the histogram's summary page for more information. You might want to collect the data again.

Message: CANFINEVE, cannot find event name '*event-name*'

Type: Warning

Explanation: The Analyzer cannot find *event-name* as an EVENT marker name.

User action: Enter a LIST /EVENT_NAMES command to determine the entire set of events. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: CANFINFIL, cannot find file name '*file-name*'

Type: Warning

Explanation: The Analyzer cannot find *file-name* as a file name in the collected data.

User action: Enter a LIST /FILE_NAMES command to determine the entire set of file names.

Message: CANMATNOT, cancel match not found

Type: Informational

Explanation: CANCEL EVENT issues this message when it does not find what you specified to be canceled.

User action: Within the Collector, use SHOW EVENT to find the events that can be canceled and reenter the command.

Message: CANOPNIPR, Cannot open IPROBE data file

Type: Error

Explanation: PCA cannot open the datafile created by IPROBE.

Message: CANNOTEXC, cannot exclude datakind

Type: Error

Explanation: Within the Analyzer, the very first datakind on the plot cannot be removed with the EXCLUDE command.

User action: See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for details. Reenter the PLOT command with the desired datakind.

Message: CANNOTSEL, cannot select requested line

Type: Warning

Explanation: PCA cannot select the requested line because there is no corresponding line DST information. This occurs because the line is recognized as a comment and not a line of machine code.

User action: Select a different line.

Message: CANTDECODE, branch/jump target at PC '*program-location*' cannot be decoded before runtime in module '*module-name*'

Type: Informational

Explanation: When a machine instruction that branches uses a variable in the destination operand, the Collector cannot determine the location for all possible next instructions. The Collector needs to determine all possible destinations so all code paths within the module can be determined. For all languages except MACRO, there is DST information to explain this. For MACRO, there is no information and so this message must be issued. Note that the Collector continues to process the code paths, but some code paths might be lost.

User action: Using the PC given with a link map, find the offending instruction and take any appropriate action. See the BADOPCODE message.

Message: CMDISOBS, command '*command-text*' is obsolete

Type: Warning

Explanation: The obsolete command is not supported by this version of PCA.

User action: None.

Message: COLNAMNOT, collection name '*collection-name*' not found

Type: Warning

Explanation: The Analyzer cannot find *collection-name* as a collection name.

User action: Enter the SHOW RUN_DESCRIPTION * command to find all the collection names.

Message: COLNUMOUT, collection number(s) out of range

Type: Warning

Explanation: For *n* collections in the data file, each Collection is numbered sequentially from 1 to *n*. If the requested collection number is greater than *n*, the Analyzer will display this message.

User action: Enter the SHOW RUN_DESCRIPTION * command to find all the collection run numbers.

Message: COLSETRES, collection settings being reset

Type: Informational

Explanation: Within the Collector, the SET and CANCEL DATAFILE commands issue this message if they cause any previously SET measurement commands to be reset to their default values. When executing either of these two commands, the Collector will always attempt to establish the original default environment.

User action: Reset any measurements to be collected that you want to remain in effect.

Message: COMNOTDEF, command '*command-text*' not defined

Type: Informational

Explanation: DELETE/COMMAND issues this message when the command you request to delete was not defined.

User action: Use SHOW SYMBOL/DEFINED to display all defined commands.

Message: CONFLICT, illegal combination of command elements

Type: Warning

Explanation: This message is issued when there exists either conflicting qualifiers or keywords within the same command.

User action: Verify the intention of your command and see if the use of a specific qualifier or keyword might mutually exclude the use of others.

Message: CONKEYATR, conflicting key attributes '*key-attribute*' and '*key-attribute*'

Type: Warning

Explanation: The recently issued DEFINE KEY command has conflicting qualifiers such as / NOSTATE and /LOCK_STATE.

User action: Determine correct applicability and reenter the command.

Message: CONQUAL, conflicting qualifiers '*qualifier*' and '*qualifier*'

Type: Warning

Explanation: The above mentioned qualifiers are mutually exclusive.

User action: Verify the intention of your command and see if the use of a specific qualifier might mutually exclude the use of other qualifiers.

Message: CPUDISTOR, CPU sampling data may be distorted by collection of other data

Type: Informational

Explanation: During the collection of CPU sampling data, the current program counter is measured at an interval of 10 milliseconds of virtual CPU time. If the Collector is also gathering other data, it will consume part of this interval to collect them. Thus, the CPU sampling measurements were distorted by the Collector.

User action: If you do not want to have your CPU sampling measurement distorted, cancel either CPU sampling or the other measurements.

Message: CREFILE, creating file '*file-name*'

Type: Informational

Explanation: The FILE command issues this message to indicate the actual file being created. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

User action: Verify that the file name in the message is what you intended.

Message: CURFILE, there is no current default listing file

Type: Error

Explanation: The APPEND command cannot find a default output file. The default output file is taken from the most recent APPEND or FILE command.

User action: Either reenter the APPEND command with a file specification as a parameter, or enter the FILE command to create a new file. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: CVTFORERR, performance data file '*file_name*' corrupt,coverage table format error

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new data file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: DATADEFPC, defaulting to collecting PC sampling data

Type: Informational

Explanation: The PC sampling measurement is selected when the user does not specify any data to be collected.

User action: If you did not desire PC sampling, stop the collection run and restart the whole process. After you are back in the Collector, set the appropriate measurements before issuing the GO command again. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information on how to set other measurements.

Message: DATARECLOST, some data records were lost at collection time

Type: Informational

Explanation: The data records used for this plot or tabulation might be incomplete.

User action: You might want to collect the data again, although the problem is likely to recur.

Message: DEFDATFIL, set datafile required in this context, creating '*file-name*'

Type: Informational

Explanation: In the Collector, many commands require symbolic information to determine if the command is valid. PCA sets the default datafile to get that information from the image being measured. The default file name is the image name appended with a .PCA file type. Note that this command also occurs after issuing the GO command without setting the data file.

User action: If the data file is not set to the one desired, enter either a CANCEL or SET DATAFILE command. Then, reissue all previously SET measurement commands. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: DEFKEY, '*state-name*' key '*key-name*' has been defined

Type: Informational

Explanation: Issued by DEFINE/KEY to inform the user that the specific key definition has been bound.

User action: If you did not want to see this message, enter /NOLOG with your DEFINE command. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: DEFKEYERR, error defining '*state-name*' key '*key-name*'

Type: Warning

Explanation: PCA uses the screen management facilities, SMG, for the purpose of managing key bindings. In this instance, the call to SMG$ADD_KEY_DEF returned a bad status.

User action: First, try to interpret the reason for the SMG error. If it can be understood, try to fix it locally. Otherwise, submit a Software Performance Report (SPR).

Message: DELKEY, '*state-name*' key '*key-name*' has been deleted

Type: Informational

Explanation: Issued by DELETE/KEY to inform the user that the specific key definition has been deleted.

User action: If you do not want to see this message, enter /NOLOG with your DELETE command.

Message: DELKEYERR, error in processing DELETE/KEY command

Type: Warning

Explanation: PCA uses the screen management facilities, SMG, for the purpose of managing key bindings. In this instance, a call to either SMG$LIST_KEY_DEFS or SMG$DELETE_KEY_DEF has returned a bad status.

User action: First, try to interpret the reason for the SMG error. If it can be understood, try to fix it locally. Otherwise, submit a Software Performance Report (SPR).

Message: DISLDRINFO, Discarded information for Lost Data Record

Sequence number = '*sequence-number*'

Sequence number of Lost Data = '*sequence-number*'

PDR of Lost Data = '*pdr-type*'

Type: Warning

Explanation: The mechanism for reporting lost data is currently overloaded and the lost data will not be recorded in the PCA data file.

User action: Some data will have been lost during collection, record the reported lost data for reference in addition to checking for missing PDRs.

Message: DISNOTSEL, display not selected because it has been removed from the screen

Type: Error

Explanation: The requested display was not selected because it was removed from the screen. Selected displays must be active.

User action: Enter a different SELECT command. Use the SHOW DISPLAY/ALL command to see the currently active displays. For more information on using the SELECT command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: DISPEXISTS, display '*display-name*' exists already; cannot be set unless canceled

Type: Error

Explanation: Each display must have a unique name. Therefore, both the SET DISPLAY and the SAVE commands must not use display names that are currently in use.

User action: Reenter the command with a display name that does not exist. To see all currently defined displays, enter the SHOW DISPLAY/ALL command. For more information on the SET DISPLAY or SAVE command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: DISPRLENSIZ, length of '*display_name*' display cannot exceed maximum size;increase /SIZE or specify fewer rows within the window

Type: Error

Explanation: The size of the display must not be less than the number of rows in the window being used.

User action: Use the SHOW WINDOW command to determine the number of rows in the window. Then with the DISPLAY command, you must either reset the size of the display with the /SIZE qualifier, or choose a different window to place the display.

Message: DISSSSINFO, Discarded information for matching Start System Service Records

Index = '*service-index*' ('*service-name*')

Link ID = '*link-identifier*'

Start CPU time = '*cpu-time*' millisec

Type: Warning

Explanation: Information for matching start system service records with IOS and SER PDRs was discarded before the required number of matches. Some data might have been lost during collection.

User action: Check for occurrences of lost IOS and SER PDRs.

Message: DOMNAMNOT, domain name '*domain-name*' not allowed as a node specification

Type: Warning

Explanation: The *domain-name* is not allowed as a node specification. The domain name specifies an entire range, whereas a node specification specifies a set within this range.

User action: Within the Collector, you must specify the set of operations to be measured. Within the Analyzer, you must specify a set of buckets to be filled. Either one unit from the respective domain must be selected, or a BY clause must be used to define the granularity. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: DWERR, DECwindows error - '*ERROR_MSG_DW*'

Type: Informational

Explanation: This is a DECwindows error that PCA cannot handle.

User action: Make sure your DECwindows environment is defined properly.

Message: DW_NOLINEBYLINE, Go to source for a source line is not supported.

Type: Warning

Explanation: PCA cannot go to the source when only a source line is selected.

User action: Go to the source by selecting a module or routine name.

Message: DW_NOPROCID, Process ID must be specified.

Type: Informational

Explanation: A process ID must be specified when merging an SPM data file.

User action: Enter a process ID in the correct OpenVMS format.

Message: DW_NOSOURCE, No source line available for selected entry.

Type: Warning

Explanation: PCA cannot go to the source for the data item selected.

Message: EDITDISVER, the displayed version is '*file-name*'

Type: Informational

Explanation: EDIT issues this message when PCA finds that the source version number in the DST is not the same as the version of the file it has found to edit.

User action: If the file name displayed is not the file you want to edit, exit the editor and PCA to edit the appropriate file. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information.

Message: EDITFILE, editing file '*file-name*'

Type: Informational

Explanation: This message informs you of the current file to be edited.

User action: If the file name displayed is not the file you want to edit, exit the editor and PCA to edit the appropriate file. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information.

Message: EDITNOFILE, no source file to use for editing

Type: Informational

Explanation: Issued by the EDIT command when the source file cannot be found.

User action: Use the SET SOURCE/EDIT command to determine whether the file has been moved. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information.

Message: EDITREVVER, editing a revised version of the displayed source file

Type: Informational

Explanation: EDIT issues this message when PCA finds that the source version number in the DST is not the same as the version of the file it has found to edit.

User action: If the file name displayed is not the file you want to edit, exit the editor and PCA to edit the appropriate file. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information.

Message: ENDCOL, data collection ends

Type: Informational

Explanation: Informational message issued by the Collector after it has stopped gathering data.

User action: If you do not want to see this message, define the logical PCA$INHIBIT_MSG.

Message: ENTRYUSRNOTFOU, Global symbol for entry routine not found in user routine image

Image = '*image-name*'

Entry User Routine = '*user-routine*'

Type: Warning

Explanation: The user routine image does not contain the specified global symbol.

User action: Check the user routine image for compliance.

Message: ERRASSIGN, the attempt to acquire an I/O channel for PCA failed

Type: Informational

Explanation: A bad status was returned from a SYS$ASSIGN service call. PCA needs to acquire I/O channels to do I/O. In this case, PCA failed to acquire such a channel.

User action: Check the quotas for your process. If this problem cannot be resolved locally, submit a Software Performance Report (SPR).

Message: ERRCRELNM, unable to create a logical name

Type: Informational

Explanation: PCA creates logical names for input and output redirection.

User action: Check the quotas for your process. If this problem cannot be resolved locally, submit a Software Performance Report (SPR).

Message: ERRDEASSIGN, attempt to deassign an I/O channel acquired by PCA failed

Type: Informational

Explanation: PCA wanted to deassign an I/O channel that was acquired. This error notes the failure of the SYS$DASSGN system service, which is probably due to an invalid channel.

User action: Submit a Software Performance Report (SPR).

Message: ERRFAO, unable to format output string

Type: Informational

Explanation: An error was returned from a call to SYS$FAO.

User action: Submit a Software Performance Report (SPR).

Message: ERRPARUSRIMG, Error parsing user routine image '*file-name*'

Type: Error

Explanation: An error occurred while parsing the specified user routine image.

User action: Make sure the user routine image specified is a valid OpenVMS file specification.

Message: ERRPROPDF, error processing performance data file '*file-name*'

Type: Error

Explanation: Either the Analyzer or Collector was unable to open the specified data file. This might result from anything that could cause an error in an RMS $OPEN or $CREATE.

User action: If a new file is to be created, verify that the file does not exist. If an old file is to be read, verify that the file does exist. If the file to be read exists, verify that it was created by the Collector. If a mistake was made, reenter the SET DATAFILE command in either the Collector or the Analyzer.

Message: ERRPROSDF, error processing shareable DST file '*file-name*'

Type: Error

Explanation: An error was detected in processing the image that contained the DSTs for the shareable image being measured. This message is issued by an obsolete form of the SET DATAFILE/ SHAREABLE command. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for more information.

Message: ERRQIOW, error from SYS$QIOW

Type: Informational

Explanation: A bad status was returned from a call to SYS$QIOW.

User action: Submit a Software Performance Report (SPR).

Message: ERRSEAUSRIMG, Error searching for user routine image '*file-name*'

Type: Error

Explanation: An error occurred while searching for the specified user routine image.

User action: Check that the location of the user routine image specified is either by default in SYS $SHARE or fully defined by a logical name.

Message: ERRSMG, error returned from a call to the Screen Management Facility (SMG)

Type: Informational

Explanation: A bad status was returned from a call to a SMG routine. This might not be a PCA problem.

User action: Check for potential interactions between the user program and PCA (pasteboard sharing and so on). In addition, check the setup of the terminal, which might cause SMG some problems. If the error still cannot be explained, submit a Software Performance Report (SPR).

Message: ERRWRIFIL, error writing file '*file-name*'

Type: Error

Explanation: The output could not be written to the appropriate file.

User action: Determine if there is any external interference with RMS writing to the specified file. If no interference can be found, submit a Software Performance Report (SPR).

Message: EXARANGE, hexadecimal number not valid; valid range is '*number*' to '*number*'

Type: Warning

Explanation: Illegal hexadecimal range provided as input.

User action: Verify that the provided hexadecimal range is in an accepted format.

Message: EXCDATAKIND, cannot include another datakind, the maximum is 8

Type: Error

Explanation: In the Analyzer, there is a maximum number of eight datakinds that can be included in a plot.

User action: If you want to add another datakind, exclude one datakind first and then include the appropriate datakind. For more information about using the INCLUDE and EXCLUDE commands, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: EXITUSRNOTFOU, Global symbol for exit routine not found in user routine image

Image = '*image-name*'

Exit User Routine = '*user-routine*'

Type: Informational

Explanation: The user routine image does not contain the specified global symbol.

User action: Check the user routine image for compliance.

Message: FILEUNAL, file not available

Type: Warning

Explanation: This message is issued when the data file is unavailable for writing because it is locked by another user.

User action: Determine which other user or process is locking your data file. Then, resolve the problem appropriately.

Message: FILNAMREQ, file name required

Type: Warning

Explanation: The FILE_NAME filter specification requires a name of a file that was measured by IO_SERVICES. With the SET FILTER command, the FILE_NAME must be set to value following standard OpenVMS file-specification conventions.

User action: Use the LIST/FILE_NAME command to determine all the files that have been measured. For more information about the SET FILTER and LIST command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: FILNOTFOU, file is not found; it must exist before appending

Type: Error

Explanation: The APPEND command expects to find a previously created file before appending.

User action: Enter the FILE command with a file name of a file to be created, or the APPEND command with a file name of an existing file. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: FILQUE, print file queued to '*queue-name*'

Type: Informational

Explanation: In the Analyzer, this message informs you where the file was queued for printing.

User action: Ensure that the file is placed on the appropriate queue. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: FIRNUMMAY, first number of a range may not be less than '*number*'

Type: Warning

Explanation: Some integers provided in the command line must be greater than a specified *n*. For example, /STACK_DEPTH= *n* must be greater than or equal to 1. The ATTACH command could not continue because a failed service call to SYS$GETJPI occurred.

User action: Determine if there is any external interference. If no interference can be found, submit a Software Performance Report (SPR).

Message: HEIGHTDIFF, desired height of '*specified_height*' is not allowed, height is set to '*actual_height*'

Type: Informational

Explanation: The inappropriate height was set in screen mode. The height of either the pasteboard or terminal will be chosen.

Message: IDENTLONG, identifier too long, please shorten

Type: Error

Explanation: The maximum size of an identifier is 255 characters. PCA will not handle program identifiers that are greater than 255. Note that a path name is composed of one or more identifiers.

Message: IDENTZERO, identifier is zero length

Type: Error

Explanation: An identifier consists of one or more legal characters in either the quoted or unquoted path-name identifier format. Note that a path name is composed of one or more identifiers.

Message: ILLEGALFILL, the fill character string is incorrect

Type: Error

Explanation: The maximum size of a fill character string is 80 characters.

User action: Reenter either the PLOT, INCLUDE, or the SET PLOT command with the appropriate fill character string in the /FILL qualifier.

Message: ILLEGALQUAL, illegal application of qualifier

Type: Error

Explanation: Not all filter specifications are allowed to have qualifiers. This message is generated when the qualifier is not allowed.

User action: Reenter the SET FILTER command without the qualifiers associated with that filter specification.

Message: ILLFORUSE, ill-formed user defined name

Type: Warning

Explanation: In the DEFINE/COMMAND command, the provided user-defined name is illegal.

User action: Reenter the command with the correct user-defined name.

Message: ILLPATHELEM, illegal pathname element at '*text*'

Type: Error

Explanation: A path name consists of identifiers delimited by a backslash (\). PCA has detected some violation of path-name rules.

User action: Correct the command and reissue it. For more information about expressing the correct syntax of the path name, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: IMGFILMIS, image and performance datafile mismatch

Type: Error

Explanation: In the Collector, the image whose performance is being measured is not the same image used to create the currently opened performance data file. In the Analyzer, the MERGE command tried to merge data from two different images. PCA uses the image name, image ID, link time, and link ID in the image header to ensure that the images are the same.

User action: Reenter the SET DATAFILE command in the Collector, or the MERGE command in the Analyzer. For more information about the SET DATAFILE command and the MERGE command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INCDST, incorrect DST nesting in module '*module-name*'. Datafile not set.

Type: Error

Explanation: PCA has detected an invalid nesting in the DEBUG symbol table for the module *module-name*.

User action: Compile the offending module *module-name* /NODEBUG, relink, and rerun the program.

Message: INCDSTNES, incorrect DST nesting in module '*module-name*' at '*program-address*'; Entry '*entry-name*'

Type: Informational

Explanation: PCA has detected invalid nesting in the Debug symbol table for *module-name*.

User action: If the OpenVMS Debugger can successfully SET MODULE *module-name*, submit a Software Performance Report (SPR). Compile the offending module with /NODEBUG, relink, and rerun the program.

Message: INCOMPSYM, symbol '*symbol*' incompatible with keyword

Type: Warning

Explanation: *Symbol* was found, but it is not in the domain specified by the keyword in the node specification. For example, you might have entered the node specification *IO_SERVICE ROUT_X*. There might be a routine ROUT_X, but it does not exist within the IO_SERVICE domain.

User action: Determine the appropriate domain for the symbol and reenter the command. For more information about node specification parsing, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INPREADERR, error reading input line

Type: Warning

Explanation: An error was found in reading the input line in the Collector. All command-line information is read from the logical PCAC$INPUT. Something is interfering with the reading from the object pointed to by this logical.

User action: Determine where PCAC$INPUT is pointing. If you cannot determine what is causing the interference, submit a Software Performance Report (SPR).

Message: INTERR, internal PCA error in '*PCA-internal-identifier*'

Type: Fatal

Explanation: PCA has detected an internal inconsistency.

User action: Submit a Software Performance Report (SPR).

Message: INTMEMERR, internal memory-pool error at '*hex-virtual-address*' '*PCA-internal-identifier*'

Type: Fatal

Explanation: PCA has detected an inconsistency in its internal memory pool.

User action: Submit a Software Performance Report (SPR).

Message: INTTOOBIG, integer too large

Type: Warning

Explanation: Integer values entered from the command line must fit into a longword.

User action: Determine if it is necessary to provide such a large number. If it is necessary, this is a restriction of PCA.

Message: INVANCNODSP, '*node specification*' is not valid for SET or CANCEL ANC

Type: Informational

Explanation: An invalid node specification was used on the SET ANC or CANCEL ANC command. For more information on the SET ANC and CANCEL ANC commands, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide and the Command Dictionary.

Message: INVARG, invalid argument

Type: Fatal

Explanation: The PCA privileged code has detected invalid arguments.

User action: Submit a Software Performance Report (SPR).

Message: INVCPTSEN, performance data file '*file-name*' corrupt,invalid code path table sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVCRTSEN, performance data file '*file-name*' corrupt,invalid collection run table sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVCVTSEN, performance data file '*file-name*' corrupt,invalid coverage table sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVDIGDEC, invalid decimal number: '*number*'

Type: Error

Explanation: A decimal digit is often expected in many contexts such as a parameter. If the provided input is not a digit, this error message will be generated.

User action: Reenter the command with the appropriate number.

Message: INVDIRNAM, invalid directory name: '*text*'

Type: Error

Explanation: Some commands expect parameters that are directory names. The provided parameter must follow standard OpenVMS conventions.

User action: Reenter the command with the appropriate directory using the standard OpenVMS format.

Message: INVDMTSEN, performance data file '*file-name*' corrupt, invalid DMT sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new data file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVGSTSEN, performance data file '*file-name*' corrupt, invalid GST sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new data file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVDSPSIZ, invalid display size: '*number*'

Type: Error

Explanation: The size of a display must be between 1 and 1000, inclusively.

User action: Enter either the DISPLAY or SET DISPLAY command with an acceptable size. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INVDSTREC, invalid DST record. Please submit an SPR.

Type: Error

Explanation: PCA has detected a DST record that it cannot understand.

User action: Submit a Software Performance Report (SPR). Be sure to include the compiler name and version number, as well as the program that generated the DST. As a workaround, compile the offending module /NODEBUG, relink the objects, and run the program again.

Message: INVDSTSEN, performance data file '*file-name*' corrupt,invalid DST sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new data file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVEVNAM, invalid event name

Type: Informational

Explanation: Event names can contain letters, numerals, underscores and dollar signs. They cannot start with numerals or contain spaces.

User action: Reenter the event name.

Message: INVIFSSEN, performance data file '*file-name*' corrupt,invalid image file spec. sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVKEYNAM, invalid or missing key name

Type: Warning

Explanation: The name of a key on the keyboard was expected in this command.

User action: Reenter the command with an acceptable name of a key. See the description of the DEFINE/KEY command in the Command Dictionary for a table of key identifiers.

Message: INVLNKPDR, Invalid PDR for link record

Sequence number = '*sequence-number*'

Link ID = '*link-identifier*'

PDR = '*pdr-type*'

Type: Error

Explanation: The PDR specified with the link identifier is not a valid linked PDR. The data in the PCA data file might have been corrupted.

Message: INVMCPSEN, performance data file '*file-name*' corrupt,invalid module code path table sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVNLSSEN, performance data file '*file-name*' corrupt,invalid collection name list sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVNUMRNG, number not valid; valid range is '*number*' to' *number*', '*number*' is taken

Type: Informational

Explanation: Some numerical input is bound by specific values. These values are specified within this message. To simplify matters, the value is chosen.

User action: Verify that the chosen value is acceptable. If not, reenter the command with a value within the accepted range.

Message: INVOPSYM, invalid operator symbol in pathname

Type: Error

Explanation: For a path name, the only valid symbol to delimit a token is a backslash (\). Any other symbol that does not belong to any identifier will cause this message to be generated.

User action: Reenter the command with the appropriate characters in the path name. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INVPAGE, invalid screen height, value must be between '*minimum_height*' and '*maximum_height*'

Type: Error

Explanation: For the SET TERMINAL command, the /PAGE qualifier must provide a page length between 11 and 100, inclusively.

User action: Reenter the SET TERMINAL command with an acceptable page size.

Message: INVPCTSEN, performance data file '*file-name*' corrupt,invalid saved coverage sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVPDSSEN, performance data file '*file-name*' corrupt,invalid collection sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVSELDIS, invalid selection of display '*display-name*'; wrong display kind

Type: Error

Explanation: The SELECT command will only select default displays that are consistent with their declaration. For example, the SELECT/PLOT will select only plot displays, and SELECT/ERROR will select only output displays.

User action: Reenter the SELECT command with a consistent display kind. For more information about using the SELECT command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INVSIFSSEN, performance data file '*file-name*' corrupt,invalid shareable DST image file spec. sentinel

Type: Error

Explanation: One of the PCA internal structures within the data file is inconsistent.

User action: Try creating a new file. If the problem is reproducible, submit a Software Performance Report (SPR).

Message: INVSRCLIN, invalid source line range

Type: Warning

Explanation: In either the TYPE or SEARCH command, the description of the lines to be used are entered in the form *[modname\] n[:m\]* where *modname* is the module, *n* and *m* are integers, and *n* is less than *m*. If any of these conditions are violated, this message is displayed.

User action: Enter the command again with the appropriate source line range. For more information about using the TYPE and SEARCH commands, see the Command Dictionary and the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: INVVECCOMB, This combination of vector datakinds is not allowed.

Type: Warning

Explanation: You cannot collect VCPU_SAMPLING, VPC_SAMPLING, or VCOUNTERS during the same collection run.

User action: Turn off either the VCPU_SAMPLING, VPC_SAMPLING, or VCOUNTERS.

Message: INVWIDTH, invalid screen width, value must be between '*minimum_width*' and '*maximum_width*'

Type: Error

Explanation: For the SET TERMINAL command, the /WIDTH qualifier can provide a page length between 40 and 255, inclusively.

User action: Reenter the SET TERMINAL command with an acceptable width.

Message: INVWINPAR, invalid window parameter: '*number*'

Type: Error

Explanation: In the Analyzer and the Collector, a newly defined window is restricted to the following conventions. First, the beginning row and the number of rows after the beginning row must be less than the terminal page length. Next, the beginning column and the number of columns after the beginning column must be less than 255. If any of the conditions are violated, this message is displayed.

User action: Provide a window parameter within the acceptable boundaries. For more information on window parameters, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: IPRCMDREQ, IPROBE command required

Type: Informational

Explanation: Gathering IPROBE data requires an IPROBE command.

User action: Reissue the SET IPROBE command with an IPROBE command line. For example:

```
SET IPROBE/COMMAND="IPROBE/PC0=CYCLES"
```

Message: IPRFILVER, IPROBE data file format is incompatible

Type: Informational

Explanation: The File format number in the IPROBE data file is not one this version of PCA understands.

User action: Use a different version of IPROBE.

Message: IVLOGNAMUSR, Invalid logical name specified for user routine image

Image = '*image-name*'

Type: Warning

Explanation: An invalid logical name was specified for the user routine image.

User action: Check the logical name used to define the location of the user routine image for compliance.

Message: IVQUAL, unrecognized qualifier '*qualifier*' – check validity,spelling, and placement

Type: Warning

Explanation: An undefined qualifier was found.

User action: Verify the syntax for the command and reenter it with only the allowed qualifiers.

Message: LICFAIL, License check failure

Type: Warning

Explanation: The PCA license check has failed.

User action: Check the appropriate license, if loaded.

Message: LINEINFO, '*information*'

Type: Informational

Explanation: This informational message is used to provide relevant information to the user. It is important to read the complete message to understand the entire context.

User action: Interpret the completely provided message and verify that the pertinent information is satisfactory.

Message: LNKPDRMIS, Linked records and PDR type mismatch

Sequence number = '*sequence-number*'

Link ID = '*link-identifier*'

expected PDR = '*pdr-type*'

received PDR = '*pdr-type*'

Type: Error

Explanation: The PDR specified with the link identifier does not match that of the actual PDR. The data in the PCA data file might have been corrupted.

Message: MATQUOMIS, matching quote is missing

Type: Warning

Explanation: All quoted strings must be delimited by either two single or two double quotation marks.

User action: Reenter the command with appropriately quoted string.

Message: MAXPARM, too many parameters – reenter command with fewer parameters

Type: Warning

Explanation: In both the SHOW KEY and DELETE KEY commands, the user cannot provide both / ALL as a qualifier and a key name as a parameter.

User action: Choose between /ALL or one key in either the SHOW KEY or DELETE KEY commands.

Message: MAXSOURLINE, Max. source lines exceeded … Choose another item in range to limit your scope

Type: Informational

Explanation: The maximum number of source lines has been exceeded.

User action: Click on another item in the range to limit the number of lines necessary.

Message: MAXSYMBOLS, the max. entries in a list box is 75 ... use filtering for a different list

Type: Informational

Explanation: There is a maximum number of entries allowed to be placed into a list box. If the number of possible entries exceeds this, the first entries preceding the maximum are placed into the dialog box.

User action: Use filtering to lower the number of possible entries.

Message: MISEVENAM, missing event name on command

Type: Warning

Explanation: Both the SET EVENT and CANCEL EVENT commands require that an event name be provided.

User action: Enter either the SET EVENT or CANCEL EVENT command with an event name. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: MODNAMHAV, module names have no address

Type: Warning

Explanation: The entered SET EVENT command requires a single program address. A module has no single address associated with it. Therefore, only routines and lines can be selected as program locations in the SET EVENT command.

User action: If you meant to define an event at any time that a routine within this module is called, define the event as the set of entry points into this module.

Message: NAMTOOLONG, name is too long: '*text*'; maximum length is '*number*'

Type: Error

Explanation: In the SHOW SYMBOL command, the wildcard name has a maximum of 255 characters.

User action: Reenter the SHOW SYMBOL command again with fewer characters in the parameter. Note that you can use the wildcard (*) to represent the last n characters.

Message: NEEDMORE, unexpected end of command line

Type: Warning

Explanation: Many commands require the presence of additional parameters or qualifiers.

User action: Review the correct syntax of the corresponding command and enter it with the appropriate parameters.

Message: NOACTPLOT, no active plot or table is defined; cannot perform command

Type: Error

Explanation: Many commands require the presence of a currently active table or histogram. If neither is present, the command is not executed.

Message: NOACTPTL, no active plot, table, or list defined; cannot perform command

Type: Error

Explanation: In the Analyzer, a plot, table, or list must be present in order to perform either the APPEND, FILE, or PRINT commands. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOANCSETCAN, no ANC points set or canceled

Type: Informational

Explanation: The user entered a SET ANC or CANCEL ANC command that did not result in any changes to the currently set ANC points. Note that only a coverage point, that is, a point defined in the Coverage Table (CVT), can be set as ANC. No action is necessary.

User action: Enter the SHOW ANC command to display all the currently set ANC points. Enter the LIST/COVERAGE command to display the contents of the CVT section. For more information on the LIST command, see the Command Dictionary and the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOATTACH, attach command failed

Type: Error

Explanation: The ATTACH command has failed to attach the requested process.

User action: Verify that the requested process exists. Check the spelling of the process name.

Message: NOBUCKETS, no buckets defined by node specification list

Type: Warning

Explanation: No buckets could be found for the corresponding node specification when building the histogram. For example, if the user asked that all lines be displayed and if all the modules were compiled with /NODEBUG, there would be no buckets available. Similarly, if the user specified /MINIMUM[= *n*] and all buckets have less than *n*%, then there would be no buckets.

User action: Try the PLOT or TABULATE command again without any of the bucket selection qualifiers such as /NOZEROS, /MINIMUM, and /MAXIMUM. If this does not work, attempt to determine if the appropriate units for the domain are present. You can use LIST/DST to determine if there is line information and LIST/CPT to determine if there is code-path information.

Message: NOBUCKTAB, there is no bucket table; either a PLOT or a TAB must precede

Type: Error

Explanation: A currently active plot or table must be present before either the EXCLUDE or INCLUDE operations can be performed.

User action: Create the bucket table with either a PLOT or TABULATE command. Then, perform the INCLUDE and EXCLUDE operations. For more information on either the INCLUDE or EXCLUDE commands, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOBUCNODSPEC, no buckets found for nodespec in position '*position*'

Type: Informational

Explanation: There were no buckets for one of the node specifications in the PLOT command. If there were other provided node specifications that do have buckets, the command will be processed using these other buckets.

User action: Change the node specification in the PLOT command.

Message: NOCALLS, no active call frame '*frame-address*'; please submit SPR

Type: Informational

Explanation: An incorrect call frame was found while collecting stack PCs.

User action: If this is reproducible, submit a Software Performance Report (SPR).

Message: NOCHILDREN, there are no children of this specific node

Type: Informational

Explanation: There are no logical children of this node. Therefore, none will be displayed.

Message: NOCLOSEQ, no closing quotation mark for a quoted string

Type: Warning

Explanation: For the DEFINE/COMMAND command, the parameter that describes the command must be enclosed with either single or double quotation marks.

User action: Reenter the DEFINE/COMMAND command with the appropriate quotation marks.

Message: NOCODEPATH, no codepath information available for '*module-name*'

Type: Informational

Explanation: The BY CODEPATH clause was used in a node specification, and no code-path information for the module could be found in the data file.

User action: Run the program with SET DATAFILE/APPEND, SET COVERAGE, or SET COUNTERS MODULE foo BY CODEPATH, GO. The codepaths are recorded in the file. For an explanation on codepaths, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOCODNODSPEC, no codepaths found for nodespec in position '*position*'

Type: Informational

Explanation: There were no code paths for one of the node specifications in the PLOT command. If there were other provided node specifications that do have buckets, the command will be processed using these other buckets.

User action: Change the node specification in the PLOT command.

Message: NOCOVSET, no coverage points defined

Type: Warning

Explanation: A command referring to the coverage set has been entered, but no such section exists in the current performance data file, or the section is empty. Note that the Collector did not create CVT sections before PCA Version 2.0. Also note that even in PCA Version 2.0 or later, if SET DATAFILE/ APPEND is used on a pre-Version 2.0 file, a CVT section is not created in it.

User action: To create a CVT section, use an appropriate SET COVERAGE command in the Collector or MERGE in the Analyzer. Use the LIST/COVERAGE to see if a CVT section exists. For details on the LIST command, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODATACOL, exiting with no data collection phase

Type: Informational

Explanation: Informational message issued when the Collector is exited without any data collection. When exiting the Collector without issuing the GO command, it is assumed that you did not want any data collected.

User action: If you wanted to collect data, reenter the Collector, set the same measurements, and issue the GO command. For more information, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODATCNT, count is zero for '*datakind*'

Type: Warning

Explanation: No data points were tallied while filling the buckets, possibly because no data was gathered for the corresponding node specification, or the filters did not allow any data points to pass.

User action: You might want to verify that data was collected. To do this, use the LIST command on the respective *datakind*. For more information on the LIST command, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODATCNTINC, count is zero for '*datakind*' on this include operation

Success

In the Analyzer, no data points were tallied in the INCLUDE operation, possibly because there was no data gathered, or because no data passed the filters.

User action: You might want to verify that data was collected. To do this, use the LIST command on the respective datakind. For more information on the LIST command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODATINFIL, no '*datakind*' collected for selected collection file

Type: Warning

Explanation: The Analyzer checks to see if the requested data was gathered in the performance data file, before it reads the file.

User action: Correct the PLOT command to request data that was collected. SHOW RUN_DESCRIPTION * might help. You might want to see if the filter specification of RUN= n is set.

Message: NODATINRUN, no '*datakind*' collected for selected collection runs

Type: Warning

Explanation: The Analyzer checks to see if the requested data was gathered in the collection run before it reads the file.

User action: Correct the PLOT command to request data that was collected. SHOW RUN_DESCRIPTION * might help. You might want to see if the filter specification of RUN= *n* is set.

Message: NODCOMLIS, node specifications within a comma list must be of the same domain

Type: Warning

Explanation: All node specifications within a comma list must come from the same domain. For example, you cannot combine PROGRAM_ADDRESS BY ROUTINE and IO_SYSTEM_SERVICE BY IO_SERVICE within the same node specification list because PROGRAM_ADDRESS and IO_SYSTEM_SERVICE are in different domains.

User action: If you want a histogram for more than one domain, you must enter separate PLOT or TABULATE commands for each of the different domains.

Message: NODEFMOD, no default module to EDIT; current plot must have /SOURCE

Type: Warning

Explanation: The default file for the EDIT command is established by the PLOT/SOURCE command.

User action: Generate the desired histogram with the /SOURCE qualifier. For more information on the /SOURCE qualifier, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODIRLISM, no source directory list in effect for '*module-name*'

Type: Warning

Explanation: A source directory list for a module must be created with SET SOURCE/MODULE before CANCEL SOURCE/MODULE can be entered.

User action: Ensure that the proper module was specified. For more information on the SET SOURCE command, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NODIRLIST, no source directory list in effect

Type: Warning

Explanation: A source directory list must be created by means of SET SOURCE before CANCEL SOURCE can be entered.

Message: NODMTIMG, no DMT in image

Type: Informational

Explanation: While in the Collector, no DMT information could be found within the image. The image was probably linked with /NOTRACEBACK.

User action: Link the image with the /DEBUG qualifier.

Message: NODSTIMG, no DST in image

Type: Informational

Explanation: While in the Collector, no DST information could be found within the image. The image was probably linked with /NOTRACEBACK.

User action: Link the image with the /DEBUG qualifier.

Message: NODSTINS, no DST in image

Type: Error

Explanation: While in the Collector, no DST information could be found within the image. The image was probably linked with the /NOTRACEBACK qualifier.

User action: Link the image with the /DEBUG qualifier.

Message: NOEVENTST, stack PCs cannot be collected for event data

Type: Warning

Explanation: Collecting STACK_PC data with event data is not supported.

Message: NOEVESET, no *type-of-data* set, address information not available

Type: Informational

Explanation: The address information for EVENT, COUNTER, or COVERAGE data is not available and was not detected until after the GO command was issued. Therefore, the information is not being collected. An example of this is LINE information for MACRO32 modules.

User action: Verify that the code was compiled and linked /DEBUG and that there is DST information for the image.

Message: NOFINDPTR, no FIND pointer found

Type: Warning

Explanation: A command referring to the current position of the FIND pointer was entered, but the pointer could not be found. An appropriate FIND command should be entered first.

Message: NOFREE, no free storage available

Type: Error

Explanation: PCA has run out of virtual memory. If the message occurs in the Collector, set fewer EVENTS, COUNTERS, or COVERAGE (this works because a structure was built for each BPT), or compile some of the modules /NODEBUG.

User action: If the message occurs in the Analyzer, use fewer displays, or enter PLOT commands with fewer buckets; that is, MODULE foo BY LINE instead of PROGRAM BY LINE. Or, raise the appropriate quotas.

Message: NOGLOBSELECT, There is nothing recognized as globally selected

Type: Informational

Explanation: If nothing is selected, it cannot be pasted. If something is selected, it is not considered part of global selection.

Message: NOGSTIMG, no GST in image

Type: Informational

Explanation: While in the Collector, no GST information could be found within the image. The image was probably linked with /NOTRACEBACK.

User action: Link the image with the /DEBUG qualifier.

Message: NOIDRPDR, IDR PDRs not gathered for run '*number*'

Type: Error

Explanation: PCA has detected that there were no SSS PDRs gathered in run *number* and /SEQUENCE_NUMBERS was requested on the LIST command.

User action: If the performance data section was collected before the introduction of IDR PDRs, recollect the performance data using a collector supporting IDR PDRs. If data was lost during collection, check for occurrences of lost IDR PDRs. Otherwise, the PCA data file might have been corrupted.

Message: NOKEYPAD, unable to set up keypad definitions

Type: Informational

Explanation: PCA was unable to set up keypad definitions because an error was returned from SMG $DELETE_VIRTUAL_KEYBOAD. It uses the SMG to perform keypad management.

User action: Check the setup of the terminal that might cause SMG some problem. If the error still cannot be explained, submit a Software Performance Report (SPR).

Message: NOKEYW, qualifier name is missing – append the name to the slash

Type: Warning

Explanation: A qualifier is composed of both a slash (/) and keyword. If only a slash appears, this error message is displayed.

User action: Ensure that there are no dangling slashes in the command.

Message: NOLBLGVN, no label given in currently active plot

Type: Warning

Explanation: The first FIND command entered in the Analyzer session must contain the parameter that is the label to be found.

User action: Provide a label for the command. For more information on the FIND command, see the Command Dictionary.

Message: NOLDRPDR, LDR PDRs not gathered for run '*number*'

Type: Informational

Explanation: PCA has detected that there were no LDR PDRs gathered in run *number* and /LOST_DATA was requested on the LIST command.

User action: If the performance data section was collected before the introduction of LDR PDRs, recollect the performance data using a collector supporting LDR PDRs and check for lost data. If no data has been lost during collection, no action is necessary.

Message: NOLINECOR, cannot correlate any codepaths to a line within '*module-name*'

Type: Informational

Explanation: Although code-path information for the module was present, the Analyzer was unable to display any code-path information because it could not correlate it to a specific line within the given module. This might be caused by the fact that the module was compiled with /NODEBUG. It might also be caused by the fact that the module might be elaboration code within Ada.

User action: If the module was compiled with /NODEBUG, recompile the specific module. If the module contains only elaboration code, use the /NOSOURCE qualifier on either the PLOT or the TABULATE command.

Message: NOLINEXI, lines '*number-range*' do not exist in module '*module-name*'

Type: Warning

Explanation: There were no lines for the specified range.

User action: Enter a different range of lines for that module.

Message: NOLINEXIS, line '*number*' does not exist in module '*module-name*'

Type: Warning

Explanation: The EDIT command displays this message when the requested line does not exist within that module.

User action: Enter the EDIT command with a different line number or module name. For more information on using the EDIT command, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOLINNODSPEC, no lines found for nodespec in position '*position*'

Type: Informational

Explanation: There were no lines found for one of the node specifications in the PLOT command. If there were other provided node specifications and these do have buckets, the command will be processed using these other buckets.

User action: Change the node specification in the PLOT command.

Message: NOMATCH, no matches

Type: Warning

Explanation: When the SEARCH command cannot find this string in the source, this message is issued.

User action: Verify that both the string and the range were correct in the SEARCH command. For examples that use the SEARCH command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOMATSSS, Could not match a Start System Service Record to a Link record

Link ID = '*link-identifier*'

Type: Informational

Explanation: There was no matching start system-service record to link with IOS and SER PDRs.

User action: If the information was discarded, see DISSSSINFO. If data was lost during collection, check for occurrences of lost SSS PDRs. Otherwise, the PCA data file might have been corrupted.

Message: NOMERGESHR, cannot merge an SPM data file with a data file that has collected a shareable image data.

Type: Warning

Explanation: PCA has found that the data file to be merged is an SPM performance data file. PCA cannot merge an SPM performance data file with a data file that has previously collected data for a shareable image.

User action: Select a different data file to merge.

Message: NONCONTST, noncontiguous code for statement number detected

Type: Informational

Explanation: During the processing of PLOT/SOURCE, PCA detected that the DST gives PCA two lines with the same line number, but different address ranges. PCA assigns an address range that includes the union of both lines.

Message: NONXTLIN, there is no default next line

Type: Warning

Explanation: There are currently no default line numbers for the most recent TYPE or SEARCH command.

User action: Reenter the command with the desired module name and line number. For examples that use the TYPE and SEARCH commands, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOOCCLDISP, '*display_name*' display may not be occluded

Type: Informational

Explanation: In screen mode, certain displays (such as the prompt display) must be visible to the user.

User action: Reposition or replace the offending display.

Message: NOPAGE, no page is generated from last PLOT or TABULATE command

Type: Warning

Explanation: There are no pages to be displayed from the last PLOT or TABULATE command in the Analyzer.

User action: Determine which qualifiers on the PLOT or TABULATE command caused the removal of all information.

Message: NOPAGEFLTST, stack PCs cannot be collected for page fault data

Type: Warning

Explanation: Collecting STACK_PC data with page faults is not supported.

Message: NOPASTEPLACE, there is no destination to paste, click on a text widget

Type: Informational

Explanation: The destination to paste is always a text widget. It is defined as the last widget to receive input focus.

Message: NOPLOTDISP, the plot display is not allowed in the Collector

Type: Error

Explanation: The Collector does not allow a display with a display kind of PLOT to be created.

Message: NOPROMPT, cannot delete, remove, un-select, or change display kind of the '*display_name*' display

Type: Informational

Explanation: A prompt display must always exist so the user can perform both input and output.

Message: NOSAVPROG, cannot save a program I/O display

Type: Informational

Explanation: Because neither program input nor output is stored in PCA, the contents of the display cannot be saved.

Message: NOSCRDEV, screen mode is not supported on this device

Type: Informational

Explanation: Screen mode is only supported on VAXstations or VT100-or-higher terminals.

Message: NOSCRMODE, screen mode is not supported on this terminal;screen mode is not set

Type: Warning

Explanation: Screen mode uses SMG as the window manager. If SMG cannot handle this type of device, this message will be displayed.

User action: Determine whether the SMG RTL works on this device with other applications. If it does, submit a Software Performance Report (SPR).

Message: NOSCROLL, missing display name, no scrolling display selected

Type: Error

Explanation: If a default scrolling display does not exist, MOVE, EXPAND, and SCROLL must have a display name as the parameter.

User action: Either use the SELECT/SCROLL command to define the current default scrolling display, or provide a display name as a parameter for the command. For details on using the SELECT command and manipulating the displays, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOSCROLLDISP, '*display_name*' display may not be scrolled

Type: Informational

Explanation: The *display_name* is not allowed to be scrolled because the information in the display is not stored locally.

User action: If you want to store a history of either the error messages or your command-line input, use a display with display kind OUTPUT. You must then choose the appropriate display as the default error or input displays using either the SELECT/ERROR or SELECT/INPUT commands, respectively. For instructions on how to SELECT a specific display, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOSETTERM, the SET TERMINAL command is not supported on this terminal

Type: Warning

Explanation: SET TERMINAL is only supported on VAXstation, VT100-or-higher terminals.

Message: NOSHRIMG, no shareable image information available

Type: Informational

Explanation: The Collector could not read the system structures to get the shareable image information.

User action: Try again. If that does not work, the image activator and Collector are not interacting properly. Submit a Software Performance Report (SPR).

Message: NOSPAWN, spawn command failed

Type: Error

Explanation: A process cannot be spawned by PCA.

User action: Check your process quotas.

Message: NOSRCHSTR, search string not set

Type: Warning

Explanation: For the SEARCH command, there must be a default search string, or a search string must be provided as a parameter.

User action: Supply the appropriate search string with another SEARCH command. For examples on using the SEARCH command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOSRCLIN, no source line for address '*program-address*'

Type: Warning

Explanation: While building the bucket table, the Analyzer could not find the correct source lines.

User action: Be sure the module was compiled with /DEBUG.

Message: NOSSSPDR, SSS PDRs not gathered for run '*number*'

Type: Informational

Explanation: PCA has detected that there were no SSS PDRs gathered in run *number* and / START_TIME was requested on the LIST command.

User action: If the performance data section was collected before the introduction of SSS PDRs, recollect the performance data using a collector supporting SSS PDRs. If data might have been lost during collection, check for occurrences of lost SSS PDRs. Otherwise, the PCA data file might have been corrupted.

Message: NOSUBTREES, there are no subtrees in the current plot tree

Type: Warning

Explanation: There are no logical subtrees in the currently traversed plot. Therefore, you cannot use the SUBTREE parameter on the NEXT, BACK, FIRST, and CURRENT traverse commands.

User action: Traverse without going to other subtrees.

Message: NOSUCHDISP, no such display defined: '*display-name*'

Type: Error

Explanation: The display must exist for most display commands.

User action: Use the SHOW DISPLAY * command.

Message: NOSUCHFIL, no such filter defined: '*filter-name*'

Type: Warning

Explanation: The specific filter must exist for most filter commands.

User action: Use the SHOW FILTER * command.

Message: NOSUCHLBL, no such label found in currently active plot: '*label-name*'

Type: Warning

Explanation: When the FIND command cannot find any other matching labels, this message is displayed.

User action: Verify that the label provided in the FIND command is correct. For more information on the FIND command, see the Command Dictionary and the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOSUCHMOD, no such module name: '*module-name*'

Type: Warning

Explanation: The specified module does not exist.

User action: Use SHOW SYMBOL *.

Message: NOSUCHSYM, no such symbol '*text*' exists

Type: Informational

Explanation: No symbols matching the specified symbol specification have been found.

User action: Check the spelling of the symbol specification in question. You can also use the SHOW SYMBOL * command to display all the symbol names.

Message: NOSUCHTASK, there exists no such task; please submit an SPR

Type: Error

Explanation: The Collector could not find any information for the specified task.

User action: Submit a Software Performance Report (SPR).

Message: NOSUCHWIND, no such window defined: '*window-name*'

Type: Error

Explanation: A window must exist before you can cancel it.

User action: Use SHOW WINDOW *.

Message: NOTADAPROG, not an Ada program, tasking information cannot be gathered

Type: Informational

Explanation: Only Ada programs can have tasking information gathered because the Collector is dependent on the presence of the Ada Run-Time Library.

Message: NOTDW_COMMAND, Not allowed to execute this command from DECwindows interface.

Type: Warning

Explanation: Certain commands cannot be executed from the DECwindows interface. Thus, they are disabled.

Message: NOTINRANGE, integer value '*number*' not in valid range of '*low-bound*' to '*high-bound*'

Type: Warning

Explanation: The interpreted integer is not within the valid range.

User action: Reenter the command with the proper range.

Message: NOTORIGSRC, original version of source file not found; file used is '*file-name*'

Type: Informational

Explanation: When the compiler creates the object file, it puts the name of the source file into the object file. If PCA cannot find the exact file, it uses the SET SOURCE information to find a likely candidate. PCA issues the message when one of the following tests fails: creation data and time must be the same; file size must be the same; record format must be the same; file organization must be the same.

User action: Use the SET SOURCE command to define the right directory to find the right source file. For an explanation of the SET SOURCE command, see the Command Dictionary and the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOTPDF, '*file-name*' is not a performance data file

Type: Error

Explanation: The data file being opened must have been created by the Collector.

User action: If this data file was created by the Collector, try creating a new one. If this is reproducible, submit a Software Performance Report (SPR).

Message: NOTVECTCAPABLE, system is not Vector capable

Type: Warning

Explanation: This system is not Vector capable (no Vector hardware or VVIEF is not enabled). To use this command, your system needs to have either Vector hardware or VVIEF enabled.

Message: NOUNINODSPEC, no units found for nodespec in position '*position*'

Type: Informational

Explanation: There were no units found for one of the node specifications in the PLOT command. If there were other provided node specifications that do have buckets, the command will be processed using these other buckets.

User action: Change the node specification in the PLOT command.

Message: NOUSERROUT, No routines specified for '*image-name*'

Type: Informational

Explanation: No user routines were specified for the user-routine image.

User action: Reenter the command with at least one user routine.

Message: NOVECTOR, Vector support is not available on VMS V5.3.

Type: Informational

Explanation: VAX Vectors is available only with systems running OpenVMS VAX Version 5.4 or later. To enable the vector support, your system needs to have OpenVMS VAX Version 5.4 or later installed.

Message: NOWBPT, cannot insert breakpoint

Type: Fatal

Explanation: The Collector was not able to set a BPT instruction on one or more of the program locations specified, probably because the program's code was protected.

User action: Change the protection so the Collector can write the BPT.

Message: NOWILDFIL, file name, type, and version cannot be wildcarded

Type: Warning

Explanation: The wildcard character cannot be used in defining the source directory in the SET SOURCE command.

User action: For more information on the SET SOURCE command, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: NOWOPCO, cannot replace breakpoint with opcode

Type: Fatal

Explanation: The Collector was not able to replace the BPT it set with the original operation code, probably because the program's code has been protected at run time.

User action: Do not protect the pages at run time.

Message: NOWPROT, cannot set protection

Type: Fatal

Explanation: The Collector cannot change the protection on a page of memory to put a BPT in it.

User action: Do not protect the pages at run time.

Message: NOWRITE, invalid command – read-only access to the performance data file

Type: Warning

Explanation: The currently active performance data file is opened for shared read-only access. As a result, the MERGE, SET ANC, and CANCEL ANC commands are not allowed because they require write access. The Analyzer allows shared access to a performance data file, but only the first user who opened it is granted read and write access. Other users are granted read-only access.

User action: Try to set this performance data file again later. To check the type of access granted to the currently active data file, enter the Analyzer SHOW DATAFILE/FULL command.

Message: NULLNLS, Name List Section '*name-list-section*' is null

Type: Informational

Explanation: The data file contains a name list section that had no entries, so the running of the Analyzer might become inconsistent. This is usually caused by creating a new data file in the Collector and then not running a subsequent collection.

User action: After creating the new data file in the Collector, always run the default collection.

Message: NULNODSET, node specification '*node specification*' describes the null set

Type: Informational

Explanation: The provided node specification does not express any set of units. For example, the node specification LINE %LINE *n* BY LINE is the null set.

User action: Reenter the command with an appropriate node specification.

Message: NUMNOTALL, numbers are not allowed in the BY clause in the Collector

Type: Warning

Explanation: In the Collector, BY *n* LINES is not accepted.

User action: Enter the node specification as BY LINE.

Message: NYI, '*feature*' not yet implemented

Type: Error

Explanation: The feature is not yet implemented.

User action: Choose a different feature.

Message: ONLFILCRE, only data files created by PCA V *number*. *number* or later are supported

Type: Error

Explanation: You have attempted to open a performance data file created by a previous version of the Collector.

User action: Create a new data file with the most recent version of the Collector.

Message: ONLVER4, image must be linked with OpenVMS Version 4.0 or later

Type: Error

Explanation: The Collector does not support image files linked on OpenVMS systems prior to Version 4.0.

User action: You must relink the image.

Message: OPTERROR1, PCA's plot optimization ('*number*') has detected an error

Type: Informational

Explanation: The Analyzer attempts to use any information from the previous plot that can be applied to the current PLOT command. The Analyzer carefully checks its internal structures for consistency. In this instance, the Analyzer has found an inconsistency. The Analyzer recovers by rebuilding the bucket table from scratch.

User action: Submit a Software Performance Report (SPR).

Message: OVFLOWBUF, overflow of the input buffer, command ignored

Type: Warning

Explanation: The command line can only process records that are less than 1024 characters long.

User action: If you want more than 1024 characters for input, divide each line into records that are less than 1024 characters and use the continuation character (a hyphen) at the point of each break. If the sum of these records is greater than 32,768 characters, the complete command line cannot be processed.

Message: PAGEFLTLOST, some page fault data was lost at collection time

Type: Informational

Explanation: The page-fault data used for this plot or tabulation might be incomplete as a result of a collection problem.

User action: You might want to collect the data again. See the summary page of the current plot for more information.

Message: PAGEFLTNOTALL, page fault collection is not allowed, page fault measurement is not set

Type: Warning

Explanation: PCA only gathers page-fault data on OpenVMS VAX, or OpenVMS Alpha Version 6.1 or later.

User action: You might want to collect the data again. See the summary page of the current plot for more information.

Message: PAGTOOSMALL, page size too small or too many title lines

Type: Warning

Explanation: In the Analyzer, there cannot be less than six lines available for the buckets on a page.

User action: Either enter the command SET TERM/PAGE= *n* or remove some of the titles.

Message: PATHTOOLONG, pathname too long at ' *text*'

Type: Error

Explanation: There are too many element identifiers describing the path name. The maximum is 25.

User action: You need only the identifiers that make the path name unique. Therefore, apply only those identifiers that are necessary.

Message: PCAERR, internal PCA coding error; PCA$COLLECTOR_MAIN: '*hex-virtual-address*'

Type: Fatal

Explanation: PCA has detected an internal inconsistency.

User action: Submit a Software Performance Report (SPR).

Message: PCDISTORT, PC sampling data may be distorted by collection of other data

Type: Informational

Explanation: During collection, PC sampling is performed by collecting a program counter at regular time intervals. If the Collector is gathering other data, it will be consuming portions of this interval for its own purpose. Therefore, PC sampling will reflect the time consumed by both the Collector and the application program.

User action: If you want accurate data, cancel either PC sampling or the other measurement.

Message: PDFNOTCRE, performance datafile not created with /SHAREABLE qualifier

Type: Error

Explanation: You have attempted to SET DATAFILE/APPEND/SHAREABLE=...to a file that was not created /SHAREABLE. A data file can only contain the measurements for one image.

User action: If you want to measure both a main image and a called shareable image, you must create two separate data files. For information on measuring shareable images, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: PFDISTORT, page fault data may be distorted by collection of other data

Type: Informational

Explanation: During collection, page-fault collection is performed by collecting all possible page faults. If the Collector is gathering other data, it will cause additional page faulting for its own work. Therefore, page faults will reflect the page faulting by both the Collector and the application program.

User action: If you want accurate data, cancel either page faults or the other measurement.

Message: PROMPTCLEN, '*display_name*' display width not changed, must be full width of screen

Type: Informational

Explanation: The width of the prompt display must be the size of the screen.

Message: PROMPTOCCL, '*display_name*' display now occludes some or all of '*display_name*' display's text

Type: Informational

Explanation: The prompt display must not be occluded.

User action: You need to place *display_name* in a window that does not cover portions of the prompt display. You might also want to place the prompt display in a different window. See the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide for details on the use of the DISPLAY command.

Message: PROMPTRLEN, '*display_name*' display length not changed, must be at leasttwo lines long

Type: Informational

Explanation: The number of lines in the prompt display must be at least two.

Message: PRORUNABO, program run aborted

Type: Warning

Explanation: This might be caused because there is no line DST information and the user requested coverage or counters for PROGRAM BY LINE.

User action: Ensure the desired DST information is within the image by performing a LIST/DST command in the Analyzer. If it is not present, recompile and link your image with the /DEBUG qualifier.

Message: PRVCODFAL1, PCA has detected an internal error during the execution of privileged code

Type: Fatal

Explanation: The Collector has detected an internal error in its privileged code. The Collector recovers from such errors with some loss of data and reports them at the end of the collection run. Note that the data in the performance data file might not be reliable. In addition, PCA puts some debugging information in the data file for use by the developers.

User action: Submit a Software Performance Report (SPR) with the performance datafile and executable image, as well as any files necessary to run the image.

Message: PRVCODFAL1I, some data were lost at collection time due to a privileged code internal error

Type: Informational

Explanation: A PCA-F-PRVCODFAL1 fatal error occurred at collection time, which indicates that PCA detected an internal error in its privileged code. The Collector recovers from such errors with some loss of data and reports them at the end of the run. Note that the data in the performance data file might not be reliable. In addition, PCA puts some debugging information in the data file for use by the developers.

User action: Submit a Software Performance Report (SPR) with the performance data file and executable image, as well as any files necessary to run the image.

Message: PZONEBAD, PCA Permanent memory zone corrupt

Type: Fatal

Explanation: Corruption has been detected in the PCA permanent memory zone. Any data collected by PCA during this run is unreliable.

User action: Verify that all operations using the RTL virtual memory routines (LIB$xxx_VM_xxx) in the user program access only zones created by the user or the default zone. Pay specific attention to situations where LIB$FIND_VM_ZONE is used to locate a zone ID. In addition, ensure that stale pointers are not used for memory references after the memory has been freed. If a user programming error is not identified, submit a Software Performance Report (SPR).

Message: PZONEMIA, PCA Permanent memory zone has been deleted

Type: Fatal

Explanation: A virtual memory zone created by PCA has been deleted.

User action: Check that all calls to LIB$DELETE_VM_ZONE in the user program, including all condition and exit handlers, delete only those zones allocated to the user. Pay careful attention to situations where LIB$FIND_VM_ZONE is used to recover zone IDs. If a user programming error is not identified, submit a Software Performance Report (SPR).

Message: QUALINCDOM, datakind qualifier inconsistent with parameter domain

Type: Warning

Explanation: The datakind qualifier used must support the provided domain with the PLOT, TABULATE, and INCLUDE commands. For example, you cannot plot the PC sampling hits against the IO_SYSTEM_SERVICE BY IO_SERVICE domain because system service information is not collected with each PC sample.

User action: To determine what datakind qualifiers are consistent with the different domains, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: QUOSTREXP, quoted string expected

Type: Warning

Explanation: A quoted string was expected. This quoted string must be delimited by either single or double quotation marks.

Message: READERR, PCA input read error, forced to exit

Type: Warning

Explanation: An error was found in reading the input line in the Collector. All command-line information is read from the logical PCAC$INPUT. Something is interfering with the reading from the object pointed to by this logical.

User action: Determine where PCAC$INPUT is pointing. If you cannot determine what is causing the interference, submit a Software Performance Report (SPR).

Message: REQQUAMIS, required command qualifier missing

Type: Warning

Explanation: Both the DEFINE and DELETE commands must have a qualifier to specify the type of definition or deletion.

User action: You must use either the /KEY or /COMMAND qualifier.

Message: RETURNED, control returned to process '*process-name*'

Type: Informational

Explanation: After performing either a SPAWN or ATTACH operation, this message is generated to inform you when you have returned to the PCA process.

Message: ROUTINDISP, target of call at PC '*program-address*' cannot be decoded before run time in module '*module-name*'

Type: Informational

Explanation: When a call instruction uses a variable destination, PCA cannot determine the destination for the transfer of control. PCA needs to determine the destination so all code paths in the module can be determined. For all languages except MACRO, it is assumed that any destination of this type has DST information and the information will not be lost. In this case, PCA will not issue the message. For MACRO, this assumption is not made and the message is issued. Note that PCA continues to process the code paths, but some code paths might be lost.

User action: Using the given PC and a link map, find the offending instruction, and take any appropriate action.

Message: SCRTOBIG, screen too big for Screen Mode, width must be less than 256, height less than 100

Type: Error

Explanation: Many of the data structures that support screen mode are dependent on these boundaries.

User action: You can use the SET TERM/PAGE= *n*/WIDTH= *m* command to set reasonable boundaries.

Message: SCRTOSMALL, screen too small for Screen Mode, width must be at least 40, height must be at least 11

Type: Error

Explanation: Many of the data structures that support screen mode are dependent on these boundaries.

User action: In both the Collector and the Analyzer, you can use the SET TERM/PAGE= *n*/WIDTH= *m* command to set reasonable boundaries.

Message: SELEXPRLONG, the text for the selection expression is too long

Type: Informational

Explanation: The maximum amount of text for the selection expression is too long.

User action: You must shrink the text into a manageable subexpression.

Message: SEQPDRMIS, Sequence number and PDR type mismatch

Sequence number = '*sequence-number*'

expected PDR = '*pdr-type*'

received PDR = '*pdr-type*'

Type: Error

Explanation: The PDR specified with the sequence number does not match that of the actual PDR. The data in the PCA data file might have been corrupted.

Message: SETDATREQ, set data file required in this context

Type: Warning

Explanation: Some commands require that a data file be set.

User action: Enter the SET DATAFILE command in the Analyzer.

Message: SETKEY, keypad state has been set to '*state-name*'

Type: Informational

Explanation: For the SET KEY operation, this informs the user that the key was set.

User action: If you do not want to see this information, use the /NOLOG qualifier on the SET KEY command.

Message: SETKEYERR, error in processing SET KEY command

Type: Warning

Explanation: The screen management facility is used to maintain screen bindings. In this instance, SMG$SET_DEFAULT_STATE has returned an error.

User action: See the SMG documentation to determine the problem. If your answer cannot be found, submit a Software Performance Report (SPR).

Message: SHOKEYERR, error in processing SHOW KEY command

Type: Warning

Explanation: The screen management facility is used to maintain screen bindings. In this instance, SMG$LIST_KEY_DEFS has returned an error.

User action: See the SMG documentation to determine the problem. If your answer cannot be found, submit a Software Performance Report (SPR).

Message: SHRNOTOPEN, cannot open shareable image '*shareable-image-name*'

Type: Error

Explanation: The shareable image specified is in the list of shareable images known by the image activator, but there is no open channel to the image (that is, it cannot be activated by LIB$FIND_IMAGE_SYMBOL), and therefore the Collector cannot access the image information required for collection. None.

Message: SHRNOTUSE, shareable image '*shareable-image-name*' not used

Type: Error

Explanation: The shareable image specified is not in the list of shareable images known by the image activator. If a shareable image is activated with LIB$FIND_IMAGE_SYMBOL, the Collector cannot measure the performance unless the image is linked with the program.

User action: Relink the main program with the image to be measured.

Message: SPAWNED, subprocess spawned

Type: Informational

Explanation: Issued when the /NOWAIT qualifier is used.

Message: SRCLINNOT, source lines not available for module '*module-name*'

Type: Warning

Explanation: When referencing a specific module, the symbol information in the image points to a specific file. If that file is not present in the defined location, this message is generated.

User action: Use the SET SOURCE command to define the proper location of the source. For an explanation of the SET SOURCE command, see the Command Dictionary and the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: STACKPCLOST, some stack PC data was lost at collection time

Type: Informational

Explanation: The stack PC data used for this plot or tabulation might be incomplete.

User action: You might want to collect the data again, although the problem is likely to recur. In the case of stack truncation, nothing can be done about it (the Collector has a limit on the number of stack call frames per data point). In case of stack corruption, some interference in stack usage probably occurs between your program and the Collector. You might want to analyze your program, however, to make sure that it is not corrupting the stack. See the histogram's summary page for more information.

Message: STANOTGAT, stack PCs not gathered

Type: Informational

Explanation: The Analyzer has detected that no stack PCs were gathered for the respective datakind in the current SET FILTER command, but stack analysis was requested.

User action: Either you should not use the /MAIN_IMAGE, /STACK_DEPTH, and /CUMULATIVE qualifiers, or you should redo the entire collection with the SET STACK_PCS command.

Message: STANOTGATF, stack PCs not gathered for the selected file

Type: Warning

Explanation: The Analyzer has detected that no stack PCs were gathered for the respective datakind in the current PLOT command, but stack analysis was requested.

User action: Either you should not use the /MAIN_IMAGE, /STACK_DEPTH, and /CUMULATIVE qualifiers, or you should redo the entire collection with the SET STACK_PCS command.

Message: STANOTGATR, stack PCs not gathered for run '*number*'

Type: Informational

Explanation: PCA has detected that there were no stack PCs gathered in run *number,* but stack analysis was requested.

User action: Either you should realize that the run was ignored during the PLOT command, or you should redo the entire collection with the SET STACK_PCS command.

Message: STGTRUNC, string truncated

Type: Informational

Explanation: PCA has attempted to output a string that is longer than its internal buffer.

User action: If it can be determined that this error is not caused by a user error, submit a Software Performance Report (SPR).

Message: STKOVF, command procedures too deeply nested; limit is 16 levels

Type: Error

Explanation: PCA allows only 16 levels of nesting within the command procedures.

User action: Embed some of the command procedures within others to decrease the level of nesting to fewer than 16.

Message: STRTOOLONG, quoted string too long; must be shortened

Type: Warning

Explanation: Usually, the maximum length is bounded by the width of the terminal for the SET TITLE command, or 255 characters for the DEFINE command.

User action: Shorten the size of the string.

Message: SYMNOTFOU, symbol '*symbol*' not found

Type: Warning

Explanation: The specified symbol could not be found in the symbol information.

User action: Verify the spelling with the SHOW SYMBOL * command.

Message: SYMNOTIO, symbol '*symbol*' not in the I/O system service domain

Type: Warning

Explanation: In defining an IO_SERVICE filter specification, the symbol specified must be an I/O system service such as SYS$PUT or SYS$GET.

User action: Reenter the SET FILTER command with the correct filter specifications.

Message: SYMNOTPC, symbol '*symbol*' not in the program address domain

Type: Warning

Explanation: In defining a PROGRAM_ADDRESS filter specification or CALL_CHAIN, the symbol specified must be a symbol within your program. Use SHOW SYMBOL *.

User action: Reenter the SET FILTER command with the correct filter specifications.

Message: SYMNOTSYS, symbol '*symbol*' not in the system service domain

Type: Warning

Explanation: In defining a SERVICE filter specification, the symbol specified must be a system service such as SYS$FAO or SYS$SPAWN.

User action: Reenter the SET FILTER command with the correct filter specifications.

Message: SYMNOTTAT, symbol '*symbol*' not in the task type domain

Type: Warning

Explanation: In defining a TASK_TYPE filter specification, the symbol specified must be a task type within your program.

User action: Use the LIST/TASK_TYPE command. Reenter the SET FILTER command with the correct filter specifications.

Message: SYMNOTTID, symbol '*symbol*' not in the task domain

Type: Warning

Explanation: In defining a TASK_IDENTIFIER filter specification, the symbol specified must be a task identifier.

User action: Use the LIST/TASK_IDENTIFIER command. Reenter the SET FILTER command with the correct filter specifications.

Message: SYMNOTUNQ, symbol '*symbol*' not unique

Type: Warning

Explanation: A symbol name must be unique. Because many variables in the program can have the same identifier, PCA allows the user to specify the appropriate scoping rules by means of a path name. Each nesting level of scope is expressed as an identifier for that routine. Each identifier is separated by a backslash (\).

User action: Enter the command with a less ambiguous path name. For an explanation of path names, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: SYNERRLABEL, syntax error in %LABEL construct

Type: Error

Explanation: The syntax for a label expression is %LABEL, followed by either a quoted or unquoted path-name identifier.

Message: SYNERRLINE, syntax error in %LINE construct

Type: Error

Explanation: The syntax for a line expression is %LINE followed by an integer.

Message: SYNERRNAME, syntax error in %NAME construct

Type: Error

Explanation: The syntax for a line expression is %NAME, followed by either a quoted or unquoted path-name identifier.

Message: SYNERRPNAME, syntax error in pathname construct at or near '*text*'

Type: Error

Explanation: A path name must be a set of identifiers separated by a backslash (\).

User action: For a description of path-name syntax, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: SYNTAX, command syntax error at or near: '*text*'

Type: Warning

Explanation: All commands have a specific syntax that must be followed. Failure to do so results in this error.

User action: See the Command Dictionary for the correct syntax of your command.

Message: TASKERROR, TASKING ERROR...returned status is '*status*'

Type: Error

Explanation: The Collector had a tasking error while measuring tasking data.

User action: Submit a Software Performance Report (SPR).

Message: TASNOTALL, tasking is not allowed, tasking measurement is not set

Type: Warning

Explanation: PCA gathers tasking data only on OpenVMS Version 4.6 or later.

Message: TASNOTGAT, tasking was not gathered

Type: Informational

Explanation: PCA has detected that there is no tasking data in the currently set performance data file, but task analysis was requested.

User action: Either collect tasking data on your Ada program, or do not request analysis. For details on collecting tasking data, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: TASNOTGATF, tasking data was not gathered for the selected file

Type: Warning

Explanation: PCA detected that no tasking data is in the current performance data file, but tasking analysis was requested.

User action: Either collect tasking data on your Ada program, or do not request analysis. For details on collecting tasking data, see the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: TOOMANTIT, too many title lines; at most '*number*' allowed

Type: Warning

Explanation: The maximum number of titles for a plot or a histogram is six. Use fewer titles.

Message: TOOMANYFILL, too many fill strings were provided, the excess were ignored

Type: Informational

Explanation: The /FILL qualifier accepts at most eight character strings to be used as fill.

Message: TRAVLIMIT, cannot traverse any further in this direction

Type: Warning

Explanation: The last leaf in the traverse tree has been found in this direction. This implies that you are at the smallest leaf in the smallest subtree if you are using the NEXT command, or you are at the biggest leaf in the biggest subtree if you are using the BACK command.

Message: TZONEBAD, PCA Temporary memory zone corrupt

Type: Error

Explanation: Corruption has been detected in the PCA temporary memory zone. Any data collected by PCA during this run is unreliable.

User action: Verify that all operations using the RTL virtual memory routines (LIB$xxx_VM_xxx) in the user program access only zones created by the user or the default zone. Pay specific attention to situations where LIB$FIND_VM_ZONE is used to locate a zone ID. In addition, ensure that stale pointers are not used for memory references after the memory has been freed. If a user programming error is not identified, submit a Software Performance Report (SPR).

Message: TZONEMIA, PCA Temporary memory zone has been deleted

Type: Error

Explanation: A virtual memory zone created by PCA has been deleted.

User action: Make sure all calls to LIB$DELETE_VM_ZONE in the user program, including all condition and exit handlers, delete only those zones allocated to the user. Pay careful attention to situations where LIB$FIND_VM_ZONE is used to recover zone IDs. If a user programming error is not identified, submit a Software Performance Report (SPR).

Message: UNACLOFIL, unable to close file '*file-name*'

Type: Error

Explanation: The SYS$CLOSE system service was not performed successfully when closing the output file.

User action: Determine what could be preventing RMS from closing the file successfully. If the reason cannot be determined, submit a Software Performance Report (SPR).

Message: UNACLOLOG, unable to close log file '*file-name*'

Type: Error

Explanation: The SYS$CLOSE system service was not performed successfully when closing the log file.

User action: Determine what could be preventing RMS from closing the file successfully. If the reason cannot be determined, submit a Software Performance Report (SPR).

Message: UNACONLOG, unable to connect log file '*file-name*'

Type: Error

Explanation: The SYS$CONNECT system service was not performed successfully when connecting the log file's RAB and FAB.

User action: Determine what could be preventing RMS from connecting the file's FAB and RAB successfully. If this cannot be done, submit a Software Performance Report (SPR).

Message: UNACREFIL, unable to create file '*file-name*'

Type: Error

Explanation: The SYS$CREATE system service was not performed successfully when creating the output file.

User action: Determine what could be preventing RMS from creating the file successfully. If the reason cannot be determined, submit a Software Performance Report (SPR).

Message: UNACRELOG, unable to create log file '*file-name*'

Type: Error

Explanation: The SYS$CREATE system service was not performed successfully when creating the output file.

User action: Determine what could be preventing RMS from creating the file successfully. If the reason cannot be determined, submit a Software Performance Report (SPR).

Message: UNACTAFR, unable to activate alignment fault reporting '*status*'

Type: Error

Explanation: The Collector failed to activate alignment fault reporting. The Collector recovers by ignoring the SET UNALIGNED command.

Message: UNACTSSI, unable to activate system service interception '*status*'

Type: Error

Explanation: The Collector failed to activate system-service interception. The Collector recovers by ignoring the SET [IO_]SERVICES commands.

Message: UNACTUSRIMG, Unable to activate user routine image

Image = '*image-name*'

Type: Error

Explanation: PCA was unable to activate the specified user-routine image.

User action: Check the user-routine image for errors.

Message: UNAOPEINP, unable to open input stream '*text*'

Type: Warning

Explanation: PCA could not read the command line correctly. PCAC$INPUT is probably defined incorrectly in the Collector, or PCAA$INPUT is defined incorrectly in the Analyzer.

Message: UNAOPESCR, unable to open logical for screen output

Type: Informational

Explanation: The pasteboard could not be created with the screen management facility. There must be some interference with SYS$OUTPUT.

User action: Determine what could be interfering with SYS$OUTPUT.

Message: UNAOPNSRC, unable to open source file '*file-name*'

Type: Warning

Explanation: The source file does not exist in the location defined by the symbol table.

User action: Use the SET SOURCE command to find the appropriate source file. For more information on the SET SOURCE command, see the Command Dictionary or the VSI DECset Performance and Coverage Analyzer for OpenVMS Command-Line Interface Guide.

Message: UNAQUEPRI, unable to queue print file

Type: Error

Explanation: The PRINT command could not be executed. An error was returned from the service call SYS$SNDJBC.

User action: Determine what could be affecting the service call SYS$SNDJBC. If no reason can be determined, submit a Software Performance Report (SPR).

Message: UNAREASRC, unable to read source file '*file-name*'

Type: Warning

Explanation: If the source file does not have READ privileges in the location defined, the file cannot be read.

User action: Ensure that you have the appropriate privileges for reading the file.

Message: UNDKEY, '*state-name*' key '*key-name*' is undefined

Type: Informational

Explanation: This message informs the user that the key was undefined by either the DELETE/KEY or UNDEFINE/KEY commands.

User action: Use the /NOLOG qualifier for the command if you do not want to see the message.

Message: UNKEYW, unrecognized keyword '*text*' – check validity and spelling

Type: Warning

Explanation: The keyword provided in the command was unrecognized in this context.

User action: See the Command Dictionary for the correct syntax of the command.

Message: UNKIPRDAT, Unknown IPROBE '*datakind*', changing to EXTERNAL

Type: Warning

Explanation: IPROBE has gathered a datakind that PCA does not recognize. To allow the user access to it, PCA is changing it to EXTERNAL.

User action: Use /EXTERNAL_P0 or /EXTERNAL_P1 as appropriate for analysis.

Message: UNMATQUO, unmatched quote or missing quote at end of string

Type: Warning

Explanation: Any token to be delimited by quotation marks must have quotation marks on both sides of the token. These quotation marks can be either single or double; however, the delimiters must be the same characters.

User action: Reenter the command with the appropriate, balanced quotation marks.

Message: UNMATSSS, Unmatched Start System Service Record

Index = '*service-index*' (*service-name*')

Link ID = '*link-identifier*'

Start CPU time = '*cpu-time*' millisec

Type: Warning

Explanation: The Start system service record was not matched the required number of times. Some data might have been lost during collection.

User action: Check for occurrences of IOS and SER PDRs.

Message: VALREQ, missing qualifier or keyword value – supply all required values '*text*'

Type: Warning

Explanation: In many commands for both the Analyzer and Collector, specific keywords or qualifiers are expected.

User action: See the Command Dictionary for the correct syntax of the command.

Message: VARBASE, targets of variable base CASE at PC '*program-address*' cannot be decoded before run time in module '*module-name*'

Type: Informational

Explanation: When a CASE instruction uses a base length, PCA cannot determine the destination of the transfer of control. PCA needs to determine the destination so all code paths in the module can be determined. For all languages except MACRO, it is assumed that any destination of this type has DST information about it, and because that information will not be lost, PCA does not issue the message. For MACRO, however, this assumption is not made and the message is issued. Note that PCA continues to process the code paths, but some code paths might be lost.

User action: Using the PC given with a link map, you can find the offending instruction and take appropriate action.

Message: VARCASE, targets of variable length CASE at PC '*program-address*' cannot be decoded before run time in module '*module-name*'

Type: Informational

Explanation: When a CASE instruction uses a variable length, PCA cannot determine the destination of the transfer of control. PCA needs to determine the destination so all code paths in the module can be determined. For all languages except MACRO, it is assumed that any destination of this type has DST information about it, and because that information will not be lost, PCA does not issue the message. For MACRO, this assumption is not made and the message is issued. Note that PCA continues to process the code paths, but some code paths might be lost.

User action: Using the PC given with a link map, you can find the offending instruction and take any appropriate action.

Message: WIDTHDIFF, desired width of '*display_width*' is not allowed, width is set to '*display_width*'

Type: Informational

Explanation: The width of the terminal must be between 40 and 255, inclusive.

Message: ZONEDBGERR, error detected in zone verification routines

Type: Error

Explanation: An unexpected error was returned by a system routine called by one of the PCA zone verification routines. The subsequent messages will provide additional information about the error condition.

User action: Submit a Software Performance Report (SPR).