

Guide to VAX DEC/Test Manager

Order Number: AI-Z330C-TE

October 1986

This manual describes the concepts, commands, and features of VAX DEC/Test Manager.

Revision/Update Information: This revised manual supersedes *VAX DEC/Test Manager User/Reference Manual* (Order Number AI-Z330B-TE).

Operating System and Version: VAX/VMS Version 4.0 or higher or MicroVMS Version 4.0 or higher

Software Version: VAX DEC/Test Manager Version 2.1

**digital equipment corporation
maynard, massachusetts**

First Printing, November 1984
Revised, December 1985
Revised, October 1986

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

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright ©1984, 1985, 1986 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	
DECUS	RSTS	
DECwriter	RSX	

digital

ZK-3300

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by T_EX, the typesetting system developed by Donald E. Knuth at Stanford University. T_EX is a trademark of the American Mathematical Society.

Contents

PREFACE	xvii
----------------	-------------

SUMMARY OF TECHNICAL CHANGES	xxi
-------------------------------------	------------

CHAPTER 1 INTRODUCTION TO DEC/TEST MANAGER	1-1
1.1 PRODUCT OVERVIEW	1-1
1.1.1 Steps in Regression Testing Without DEC/Test Manager _____	1-2
1.1.2 Steps in Regression Testing with DEC/Test Manager _____	1-3
1.1.3 Getting HELP _____	1-4
1.2 GETTING STARTED	1-5
1.2.1 Invoking DEC/Test Manager _____	1-5
1.2.2 Creating and Selecting a Library _____	1-6
1.2.2.1 Creating a Library—CREATE LIBRARY • 1-7	
1.2.2.2 Selecting an Existing Library—SET LIBRARY • 1-7	
1.2.3 Creating Tests and Setting Up a Test System _____	1-8
1.2.3.1 Creating a Noninteractive Test • 1-8	
1.2.3.2 Creating an Interactive Test—CREATE TEST_DESCRIPTION/RECORD • 1-11	
1.2.4 Running Tests _____	1-15
1.2.4.1 Creating Collections—CREATE COLLECTION • 1-15	
1.2.4.2 Executing Collections • 1-15	
1.2.5 Examining Test Results—REVIEW _____	1-18
1.2.6 Exiting the Review Subsystem—EXIT _____	1-20
1.3 COMMAND SUMMARIES	1-20
1.3.1 Invoking DEC/Test Manager _____	1-21
1.3.2 Creating or Selecting a DEC/Test Manager Library _____	1-21
1.3.3 Using a DEC/Test Manager Library _____	1-22
1.3.4 Logging History Information _____	1-22
1.3.5 Creating and Maintaining Tests _____	1-23
1.3.6 Categorizing Tests with Groups _____	1-24

1.3.7	Generalizing Test-Related Files with Variables _____	1-24
1.3.8	Organizing Test-Related Files _____	1-25
1.3.9	Executing Tests _____	1-26
1.3.10	Reviewing Test Results _____	1-26
	1.3.10.1 Using the Review Subsystem • 1-27	
	1.3.10.2 Selecting Test Results to Review • 1-28	
	1.3.10.3 Manipulating Test Results • 1-28	

CHAPTER 2	GETTING STARTED WITH DEC/TEST MANAGER	2-1
------------------	--	------------

2.1	DEFINING A DEC/TEST MANAGER LIBRARY	2-1
2.2	DEFINING ACCESS TO A LIBRARY	2-4
2.2.1	UIC Protection _____	2-5
2.2.2	ACL Protection _____	2-5
2.3	CREATING A LIBRARY—CREATE LIBRARY	2-8
2.4	CONVERTING A LIBRARY—CONVERT LIBRARY	2-9
2.5	SELECTING AN EXISTING LIBRARY	2-10

CHAPTER 3	ORGANIZING A TEST SYSTEM	3-1
------------------	---------------------------------	------------

3.1	TEST DESCRIPTIONS	3-1
3.1.1	The Template File _____	3-3
3.1.2	The Benchmark File _____	3-3
3.1.3	The Test Prologue File _____	3-4
3.1.4	The Test Epilogue File _____	3-4
3.2	CREATING NONINTERACTIVE TESTS	3-4
3.2.1	Writing the Test and the Template File _____	3-5
3.2.2	Creating the Test Description—CREATE TEST_DESCRIPTION _____	3-6
3.3	CREATING INTERACTIVE TESTS	3-7

3.3.1	Recording an Interactive Test and Creating its Test Description—CREATE TEST_DESCRIPTION/RECORD _____	3-8
3.3.1.1	Functions Available While Recording an Interactive Test • 3-9	
3.3.1.2	Restrictions on Interactive Testing • 3-12	
3.3.1.3	Example Interactive Terminal Session • 3-14	
3.4	DISPLAYING BENCHMARK FILES FOR INTERACTIVE TESTS—DISPLAY/BENCHMARK	3-17
3.5	RUNNING SESSION FILES—PLAY	3-19
3.6	MODIFYING TEST DESCRIPTIONS—MODIFY TEST_DESCRIPTION	3-20
3.7	COPYING TEST DESCRIPTIONS—COPY TEST_DESCRIPTION	3-22
3.8	DISPLAYING TEST DESCRIPTIONS—SHOW TEST_DESCRIPTION	3-24
3.9	DELETING TEST DESCRIPTIONS—DELETE TEST_DESCRIPTION	3-25
3.10	STORING FILES OUTSIDE THE DEC/TEST MANAGER LIBRARY	3-26
3.10.1	Establishing Default Directories for Benchmark and Template Files—SET BENCHMARK_DIRECTORY and SET TEMPLATE_DIRECTORY _____	3-26
3.10.2	Using DEC/CMS Libraries for Benchmark and Template Files _____	3-27
3.10.3	Establishing Collection Prologue and Epilogue Files—SET PROLOGUE and SET EPILOGUE _____	3-29
3.10.4	Establishing Test Prologue and Epilogue Files—CREATE TEST_DESCRIPTION/PROLOGUE and /EPILOGUE _____	3-29
3.10.5	Storing Prologue and Epilogue Files in DEC/CMS Libraries _____	3-30

<hr/>		
CHAPTER 4	SELECTING AND RUNNING TEST COLLECTIONS	4-1
4.1	CREATING COLLECTIONS—CREATE COLLECTION	4-3
4.2	EXECUTING COLLECTIONS	4-5
4.2.1	Executing Collections in Batch—SUBMIT _____	4-7
4.2.2	Executing Collections Interactively—RUN _____	4-9
4.3	COMPARING TEST RESULTS—COMPARE	4-10
4.4	DISPLAYING THE COLLECTION SUMMARY—SHOW COLLECTION	4-12
4.5	DELETING COLLECTIONS—DELETE COLLECTION	4-13
4.6	RECREATING COLLECTIONS—RECREATE	4-14
4.7	STOPPING COLLECTIONS—STOP	4-15
<hr/>		
CHAPTER 5	EVALUATING TEST RESULTS	5-1
5.1	OVERVIEW OF THE REVIEW SUBSYSTEM	5-1
5.2	REVIEW SUBSYSTEM TERMINOLOGY	5-2
5.2.1	Result Descriptions _____	5-2
5.2.2	Output Files _____	5-3
5.2.3	Comparison Status _____	5-4
5.2.4	Identifying Result Descriptions _____	5-5
5.3	INVOKING AND USING THE REVIEW SUBSYSTEM	5-6
5.4	LOCATING TEST RESULTS	5-7
5.5	EXAMINING TEST RESULTS	5-9

5.5.1	Displaying Test Results _____	5-9
5.5.1.1	Displaying Output Files for Noninteractive Tests • 5-10	
5.5.1.2	Displaying Output Files for Interactive Tests • 5-10	
5.5.2	Printing Test Results _____	5-16
5.6	WORKING WITH TEST RESULTS	5-17
5.6.1	Creating Benchmark Files from Within the Review Subsystem _____	5-18
5.6.1.1	Updating an Existing Benchmark File • 5-18	
5.6.1.2	Creating a Benchmark File for a New Test • 5-19	
5.6.2	Creating a Group from Within the Review Subsystem _____	5-21
5.7	REVIEWING PARTIALLY RUN COLLECTIONS	5-23
5.8	LEAVING THE REVIEW SUBSYSTEM	5-24
5.9	CANCELING REVIEW SUBSYSTEM COMMANDS	5-25
5.10	USING THE REVIEW SUBSYSTEM KEYPAD	5-25
5.10.1	Default Key Definitions _____	5-25
5.10.2	Redefining the Key Definitions _____	5-27

CHAPTER 6	ENHANCING THE TEST SYSTEM	6-1
------------------	----------------------------------	------------

6.1	USING PROLOGUES AND EPILOGUES	6-1
6.1.1	Test Prologues and Epilogues _____	6-2
6.1.2	Collection Prologues and Epilogues _____	6-3
6.2	USING VARIABLES	6-3
6.2.1	Creating Variables—CREATE VARIABLE _____	6-4
6.2.2	Modifying Existing Variables—MODIFY VARIABLE _____	6-5
6.2.3	Deleting Variables—DELETE VARIABLE _____	6-6
6.2.4	Using Global Variables _____	6-6
6.2.5	Using Local Variables _____	6-7
6.2.6	Overriding Variable Default Values _____	6-7

6.3	VARIABLES DEFINED BY DEC/TEST MANAGER	6-11
6.3.1	DTM\$RESULT Logical Name _____	6-11
6.3.2	DTM\$TEST_NAME Local String Symbol _____	6-12
6.3.3	DTM\$COLLECTION_NAME Global String Symbol ____	6-13
6.4	GROUPING TESTS	6-13
6.4.1	Creating Groups—CREATE GROUP _____	6-15
6.4.2	Inserting Test Descriptions into Groups—INSERT TEST_DESCRIPTION _____	6-15
6.4.3	Inserting Groups into Other Groups—INSERT GROUP _____	6-17
6.4.4	Examining the Contents of Groups—SHOW GROUP .	6-19
6.4.5	Removing the Contents of Groups—REMOVE GROUP and REMOVE TEST_DESCRIPTION _____	6-20
6.4.6	Deleting Groups from the Library—DELETE GROUP _	6-22
6.5	USING THE KEYPADS	6-26
6.6	CORRECTING ERRORS IN THE DEC/TEST MANAGER LIBRARY—VERIFY AND VERIFY/RECOVER	6-31
6.7	USING COMMAND FILES	6-32
6.7.1	Using a DEC/Test Manager Initialization Command File _____	6-34
6.8	SPAWNING OR ATTACHING TO ANOTHER PROCESS—SPAWN AND ATTACH	6-34
6.9	USING FILTERS	6-35
6.10	HISTORY LOGGING	6-36
6.10.1	Displaying the History File—SHOW HISTORY _____	6-36
6.10.2	Entering Remarks on DEC/Test Manager Commands	6-37
6.11	DISPLAYING INFORMATION ABOUT THE CURRENT DEC/TEST MANAGER LIBRARY—SHOW ALL	6-38

CHAPTER 7 USING DEC/TEST MANAGER WITH THE VAX PERFORMANCE AND COVERAGE ANALYZER 7-1

7.1	THE BENEFITS OF USING THESE TWO TOOLS TOGETHER	7-2
7.2	USING THESE TWO TOOLS TOGETHER	7-2
7.2.1	Linking Your Program with the Collector _____	7-3
7.2.2	Creating a Collector Initialization File _____	7-3
7.2.3	Creating a DEC/Test Manager Collection Prologue File _____	7-4
7.2.4	Creating a DEC/Test Manager Collection _____	7-5
7.2.5	Executing a DEC/Test Manager Collection _____	7-6
7.2.6	Using the Analyzer from the Review Subsystem _____	7-8
7.3	A SAMPLE SESSION USING THESE TWO TOOLS TOGETHER	7-10
7.4	NOTES OF INTEREST	7-12
7.4.1	System Considerations When Using These Two Tools Together _____	7-12
7.4.2	Using the Collector SET COVERAGE Command with the /PREVIOUS Qualifier _____	7-13

COMMAND DICTIONARY

CD.1	COMMAND FORMAT	CD-1
CD.2	COMMAND PARAMETERS	CD-1
CD.2.1	Collection-Name _____	CD-2
CD.2.2	File-Spec _____	CD-2
	CD.2.2.1 Directory-Spec • CD-3	
	CD.2.2.2 File-Name • CD-3	
	CD.2.2.3 File-Type and Version • CD-3	
CD.2.3	Group-Name _____	CD-4
CD.2.4	Remark _____	CD-4
CD.2.5	Result-Description-Name _____	CD-4
CD.2.6	Test-Name _____	CD-5

CD.2.7	Variable-Name _____	CD-5
CD.2.8	Variable-Value _____	CD-5
CD.2.9	Expressions _____	CD-6
	CD.2.9.1 Collection-Expression • CD-7	
	CD.2.9.2 Group-Expression • CD-7	
	CD.2.9.3 Object-Expression • CD-7	
	CD.2.9.4 Result-Description-Expression • CD-7	
	CD.2.9.5 Test-Expression • CD-7	
	CD.2.9.6 Test-Group-Expression • CD-8	
	CD.2.9.7 Variable-Expression • CD-8	
CD.2.10	Wildcard Characters _____	CD-8
CD.3	QUALIFIERS	CD-9
CD.3.1	Command Qualifiers _____	CD-9
CD.3.2	Parameter Qualifiers _____	CD-10
CD.4	RULES FOR ENTERING AND CANCELING COMMANDS	CD-10
CD.4.1	Entering Commands _____	CD-10
CD.4.2	Abbreviating Commands _____	CD-11
CD.4.3	Canceling Commands _____	CD-11
CD.4.4	Leaving DEC/Test Manager _____	CD-11
CD.5	COMMAND EXECUTION	CD-12

COMMAND DESCRIPTIONS

(\$) DTM	CD-15
DTM @FILE-SPEC	CD-17
DTM ATTACH	CD-18
DTM COMPARE	CD-20
DTM CONVERT LIBRARY	CD-25
DTM COPY TEST_DESCRIPTION	CD-27
DTM CREATE COLLECTION	CD-31
DTM CREATE GROUP	CD-39
DTM CREATE LIBRARY	CD-41
DTM CREATE TEST_DESCRIPTION	CD-43
DTM CREATE VARIABLE	CD-51

DTM DEFINE/KEY	CD-55
DTM DELETE COLLECTION	CD-59
DTM DELETE GROUP	CD-61
DTM DELETE HISTORY	CD-63
DTM DELETE TEST_DESCRIPTION	CD-65
DTM DELETE VARIABLE	CD-68
DTM DISPLAY	CD-70
DTM EXIT	CD-72
DTM EXTRACT	CD-73
DTM HELP	CD-76
DTM INSERT GROUP	CD-77
DTM INSERT TEST_DESCRIPTION	CD-80
DTM MODIFY GROUP	CD-83
DTM MODIFY TEST_DESCRIPTION	CD-85
DTM MODIFY VARIABLE	CD-94
DTM PLAY	CD-97
DTM RECREATE	CD-99
DTM REMARK	CD-101
DTM REMOVE GROUP	CD-103
DTM REMOVE TEST_DESCRIPTION	CD-105
DTM REVIEW	CD-108
DTM_REVIEW> @FILE-SPEC	CD-116
DTM_REVIEW> ATTACH	CD-117
DTM_REVIEW> BACK	CD-119
DTM_REVIEW> DEFINE/KEY	CD-121
DTM_REVIEW> EXIT	CD-125
DTM_REVIEW> FIRST	CD-126
DTM_REVIEW> HELP	CD-127
DTM_REVIEW> INSERT	CD-128
DTM_REVIEW> LAST	CD-131
DTM_REVIEW> NEXT	CD-132
DTM_REVIEW> PCA	CD-134
DTM_REVIEW> PRINT	CD-135
DTM_REVIEW> SELECT	CD-139

DTM_REVIEW> SHOW	CD-140
DTM_REVIEW> SPAWN	CD-146
DTM_REVIEW> UPDATE	CD-150
DTM RUN	CD-153
DTM SET BENCHMARK_DIRECTORY	CD-156
DTM SET EPILOGUE	CD-158
DTM SET LIBRARY	CD-160
DTM SET PROLOGUE	CD-162
DTM SET TEMPLATE_DIRECTORY	CD-164
DTM SHOW ALL	CD-166
DTM SHOW BENCHMARK_DIRECTORY	CD-167
DTM SHOW COLLECTION	CD-168
DTM SHOW EPILOGUE	CD-171
DTM SHOW GROUP	CD-172
DTM SHOW HISTORY	CD-175
DTM SHOW LIBRARY	CD-180
DTM SHOW PROLOGUE	CD-181
DTM SHOW TEMPLATE_DIRECTORY	CD-182
DTM SHOW TEST_DESCRIPTION	CD-183
DTM SHOW VARIABLE	CD-188
DTM SHOW VERSION	CD-191
DTM SPAWN	CD-192
DTM STOP	CD-196
DTM SUBMIT	CD-198
DTM VERIFY	CD-201

APPENDIX A	DEC/TEST MANAGER MESSAGES	A-1
A.1	MESSAGE DISPLAY	A-1
A.1.1	Message Format _____	A-1
A.1.2	Severity Codes _____	A-2
A.2	THE DEC/TEST MANAGER MESSAGES	A-2

APPENDIX B SESSION FILES AND SESSION FILE EDITING **B-1**

B.1	SESSION FILES	B-2
B.1.1	Sample SESSION File _____	B-2
B.1.2	SESSION File Structure _____	B-4
	B.1.2.1 Record Structure of SESSION Files • B-6	
	B.1.2.2 Modifying Session Files Directly • B-9	
B.2	INPUT FILES	B-11
B.2.1	Sample INPUT File _____	B-11
B.2.2	Special Strings _____	B-12
	B.2.2.1 Types of Special Strings Recognized by DEC/Test Manager • B-12	
	B.2.2.2 Using Special Strings in INPUT Files • B-15	
B.3	EDITING SESSION FILES USING INPUT FILES	B-16
B.3.1	Extracting an INPUT File from a SESSION File—EXTRACT _____	B-16
B.3.2	Creating an INPUT File with a Text Editor _____	B-17
B.3.3	Recording a SESSION File from an INPUT File _____	B-18
	B.3.3.1 Using the CREATE or MODIFY TEST_DESCRIPTION/INPUT Command • B-18	
	B.3.3.2 Using the INSERT Recording Function— CTRL/P I • B-19	
	B.3.3.3 Terminal Characteristics • B-20	
	B.3.3.4 Type-Ahead • B-21	
B.4	TRANSLATION TABLES	B-21

APPENDIX C INTERACTIVE TESTING EXAMPLE **C-1**

GLOSSARY **Glossary-1**

INDEX

FIGURES

1-1	DEC/Test Manager and Regression Testing _____	1-3
2-1	The DEC/Test Manager Test System _____	2-2
2-2	DEC/Test Manager Library Directory _____	2-3
4-1	Steps in Test Execution Using DEC/Test Manager _____	4-7
5-1	Review Subsystem Default Key Definitions _____	5-26
6-1	Inserting Test Descriptions into Groups _____	6-24
6-2	Inserting Groups into Groups _____	6-25
6-3	Default Review Subsystem Keypad _____	6-29
6-4	Review SHOW/RESULT, Review SHOW/BENCHMARK, and DISPLAY/BENCHMARK Keypad _____	6-30
6-5	Review SHOW/DIFFERENCES Keypad _____	6-31
7-1	Phases in Test Execution When Using DEC/Test Manager with the VAX Performance and Coverage Analyzer _____	7-8
B-1	Format of the Terminal Characteristics Block _____	B-7

TABLES

1	New Command Qualifiers _____	xxiii
2	Modified Commands _____	xxiv
1-1	Commands to Get Started Using DEC/Test Manager _____	1-21
1-2	Commands for Creating or Selecting a DEC/Test Manager Library _____	1-21
1-3	General Function DEC/Test Manager Commands _____	1-22
1-4	History Logging Commands _____	1-23
1-5	Commands for Creating and Maintaining Tests _____	1-23
1-6	Commands for Categorizing Tests with Groups _____	1-24
1-7	Commands for Generalizing Test-Related Files with Variables _____	1-24
1-8	Commands for Organizing Test-Related Files _____	1-25
1-9	Commands for Executing Tests _____	1-26
1-10	General Review Subsystem Commands _____	1-27
1-11	Commands to Select Test Results to Review _____	1-28
1-12	Commands to Manipulate Result Descriptions _____	1-28

2-1	Access Required for Individual DEC/Test Manager Commands	2-6
3-1	Test Description Fields _____	3-2
3-2	Functions Available While Recording an Interactive Terminal Session _____	3-10
3-3	Key Definitions for DISPLAY/BENCHMARK _____	3-18
5-1	Key Definitions for SHOW/RESULT and SHOW/BENCHMARK _____	5-12
5-2	Key Definitions for SHOW/DIFFERENCE _____	5-14
6-1	Filters Provided by DEC/Test Manager _____	6-35
B-1	SESSION File Record Types _____	B-8
B-2	Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File _____	B-22
B-3	Translation of Special Strings Representing Control and Nonprinting Characters When Recording a SESSION File from an INPUT File _____	B-27
B-4	Translation of Special Strings Representing 8-Bit Control Characters When Recording a SESSION File from an INPUT File _____	B-30
B-5	Translation of Special Strings Representing the Function Key Codes When Recording a SESSION File from an INPUT File _____	B-31
B-6	Translation of Special Strings Representing the Editing Key Codes When Recording a SESSION File from an INPUT File _____	B-33
B-7	Translation of Special Strings Representing the Keypad Key Codes When Recording a SESSION File from an INPUT File _____	B-34
B-8	Translation of Special Strings Representing the Arrow Key Codes When Recording a SESSION File from an INPUT File _____	B-36
B-9	Translation of Special Strings Representing the Recording Functions When Recording a SESSION File from an INPUT File _____	B-37



Preface

Objectives

This manual explains how to use VAX DEC/Test Manager as an automated regression test system. It describes how you use DEC/Test Manager in a software development environment during both the full-scale development phase and the maintenance phase of a software development project.

DEC/Test Manager allows the software programmer flexibility in organizing tests, in selecting tests for execution, and in reviewing and verifying test results. This manual is a reference book with examples that illustrate both basic and advanced techniques for using DEC/Test Manager.

Intended Audience

This manual is intended primarily for software developers who are responsible for producing fully tested code. Using DEC/Test Manager can increase the productivity of your development projects by automating your software testing system.

Structure of This Document

The *Guide to VAX DEC/Test Manager* is divided into seven chapters, a Command Dictionary, three appendixes, and a glossary.

Chapter 1, *Introduction to DEC/Test Manager*, explains regression testing and gives an overview of how DEC/Test Manager automates the regression testing process. Examples are used to explain how to organize a test system, how to run tests, and how to evaluate test results. This chapter also summarizes the DEC/Test Manager commands by function.

Chapter 2, *Getting Started with DEC/Test Manager*, explains how to create a DEC/Test Manager library and how to convert an existing DEC/Test Manager library for use with this version of DEC/Test Manager. It also explains how to define access to the DEC/Test Manager library.

Chapter 3, *Organizing a Test System*, explains how to organize your tests using a DEC/Test Manager library. Topics include how to create both interactive and noninteractive tests, how to create and modify test descriptions, how to store benchmark and template files outside the DEC/Test Manager library, and how to categorize test descriptions into groups.

Chapter 4, *Selecting and Running Test Collections*, explains the concept of collections and how to create, select, run, delete, and compare collections.

Chapter 5, *Evaluating Test Results*, explains how to examine test results. Topics include how to evaluate the results of a test run, how to examine and update test results, and how to display and print reports of the test results.

Chapter 6, *Enhancing the Test System*, describes how to enhance your test system. Topics include local and global variables, prologues and epilogues, DEC/Test Manager-supplied symbols and logicals, the DEC/Test Manager keypads, initialization and command files, filters for test result files, history logging, and spawning subprocesses.

Chapter 7, *Using DEC/Test Manager with the VAX Performance and Coverage Analyzer*, describes how to use the VAX Performance and Coverage Analyzer with DEC/Test Manager to collect and analyze performance and coverage data gathered while tests are running.

The Command Dictionary includes the DEC/Test Manager command syntax and command line format. Detailed descriptions of all DEC/Test Manager commands are listed alphabetically by command name.

Appendix A, *DEC/Test Manager Messages*, lists and explains all DEC/Test Manager messages.

Appendix B, *SESSION Files and SESSION File Editing*, describes the format of SESSION files and INPUT files. It describes how to edit SESSION files using INPUT files.

Appendix C, *Interactive Testing Example*, contains an example of an interactive test using VAX FMS.

The glossary defines terms related to DEC/Test Manager.

Associated Documents

The following list describes additional documentation related to DEC/Test Manager:

- The *VAX DEC/Test Manager Quick Reference Guide* provides a concise summary of DEC/Test Manager rules, commands, and qualifiers.
- The *VAX DEC/Test Manager Installation Guide* supplies the instructions for installing DEC/Test Manager on a VAX/VMS operating system.
- The *VAX DEC/CMS Reference Manual* provides reference information about DEC/CMS.
- The *VAX Performance and Coverage Analyzer User's Reference Manual* provides reference information about the VAX Performance and Coverage Analyzer.

Conventions Used in This Document

The following conventions are used in this manual:

- UPPERCASE words and letters used in examples indicate that you type the word or letter exactly as shown.
- Lowercase words and letters used in examples indicate that you substitute a word or value of your choice.
- Square brackets ([]) indicate optional elements.
- A horizontal ellipsis (. . .) indicates that the preceding items can be repeated one or more times.
- A vertical ellipsis
.
.
.
in an example indicates that not all the statements are shown.
- In interactive examples, all output lines and prompting characters that the system prints or displays are shown in black letters. All characters that you type are shown in red letters.
- The symbol CTRL/x indicates that you press and hold down the key labeled CTRL and then press another key (for example, CTRL/P or CTRL/Z).
- **Boldface** words in text denote words defined in the glossary.



Summary of Technical Changes

The following list describes the functional changes and additions made to this version of DEC/Test Manager. You can now do the following:

- Edit SESSION files using intermediate INPUT files:
 - Use DEC/Test Manager to extract an INPUT file from an existing SESSION file.
 - Edit the INPUT file using the text editor of your choice.
 - Create an INPUT file from scratch using the text editor of your choice.
 - Reformat an existing test script as an INPUT file using the text editor of your choice.
 - Use DEC/Test Manager to record a SESSION file from a correctly formatted INPUT file.

- Review test results more easily:

- Invoke the Review subsystem in read-only mode to browse through the result descriptions and to print files.

DEC/Test Manager allows an unlimited number of read-only reviewers at a time. DEC/Test Manager allows only one primary reviewer at a time, who can make changes to the result descriptions.

- Specify a result-description-expression on most Review subsystem commands that previously accepted a result-description-name.
- Specify comparison status qualifiers and output file qualifiers in conjunction with a result-description-expression to review result descriptions for groups of tests with similar results.
- Specify a result-description-expression on the INSERT command.
- Specify the SHOW/FILES command to display information about result descriptions.
- Attempt to update the benchmark file for a result description which has the comparison aborted comparison status.

Summary of Technical Changes

- Use new filters:
 - Use the date filter for additional date formats: dd-mmm-yyyy, mm/dd/yy, and Month, Day, Year.
 - Use the version filter to filter version numbers on file specifications.
- Specify the /NOVERIFY qualifier on the CREATE COLLECTION command to create the collection without verifying the existence of files referenced by test descriptions in the collection.
- Specify the /NOTEMPLATE qualifier on the COPY TEST_DESCRIPTION command to cause the template file name to default to test-name.SESSION for interactive tests and to test-name.COM for noninteractive tests.
- Use the following new recording functions when recording an interactive terminal session:
 - CTRL/P CTRL/C (ABORT) aborts the recording session without saving the SESSION and benchmark files, and without creating or modifying the test description.
 - CTRL/P D (DELAY) specifies a delta time DEC/Test Manager is to wait when the terminal session being recorded is played or executed.
 - CTRL/P I (INSERT) inserts an INPUT file into the terminal session being recorded.
 - CTRL/P W (WAIT) specifies a delta time DEC/Test Manager is to wait when the terminal session being recorded is played or executed.
 - CTRL/P ! (BEGIN_COMMENT) provides a simple line editing mode to enter a comment into a SESSION file. Terminate the comment by pressing CTRL/Z.

The new commands for DEC/Test Manager Version 2.1 are EXTRACT and MODIFY GROUP.

Table 1 lists the new command qualifiers and the commands to which they apply.

Table 1: New Command Qualifiers

Qualifier	Commands
/COMPARISON_ABORTED	DTM_REVIEW> BACK DTM_REVIEW> INSERT DTM_REVIEW> NEXT DTM_REVIEW> PRINT DTM_REVIEW> SHOW
/[NO]CONFIRM	RECREATE
/FILES	DTM_REVIEW> SHOW
/FULL	SHOW GROUP
/INPUT	CREATE TEST_DESCRIPTION MODIFY TEST_DESCRIPTION
/[NO]LOG	DTM_REVIEW> PRINT DTM_REVIEW> UPDATE
/SELECTED	DTM_REVIEW> PRINT
/[NO]READ_ONLY	REVIEW
/NOTEMPLATE	COPY TEST_DESCRIPTION
/[NO]VERIFY	CREATE COLLECTION

Summary of Technical Changes

Table 2 lists the existing commands modified for DEC/Test Manager Version 2.1.

Table 2: Modified Commands

Command	Description
COMPARE	Messages generated by the COMPARE command now name the test whose comparison has completed. The COMPARE command now purges test-name.BMK_SCREEN files in DTM\$LIB whenever a new version of the file is created.
CONVERT	The CONVERT command is now the CONVERT LIBRARY command.
COPY TEST_DESCRIPTION	The COPY TEST_DESCRIPTION command now takes the /NOTEMPLATE qualifier. This qualifier causes the name for the new test description to be test-name.COM for noninteractive tests and test-name.SESSION for interactive tests.
CREATE COLLECTION	The CREATE COLLECTION command now takes the /[NO]VERIFY qualifier. This qualifier determines whether DEC/Test Manager verifies the existence of files associated with test descriptions before creating the collection. Collections created with this version of DEC/Test Manager cannot be used with earlier versions.
CREATE TEST_DESCRIPTION	The CREATE TEST_DESCRIPTION command now takes the /INPUT qualifier. This qualifier specifies that DEC/Test Manager begin an interactive terminal session and take all input from the specified INPUT file until the INPUT file is exhausted. The /FILTER qualifier can now specify a file version filter and more general date filters.

Table 2 (Cont.): Modified Commands

Command	Description
MODIFY TEST_DESCRIPTION	The MODIFY TEST_DESCRIPTION command now takes the /INPUT qualifier. This qualifier specifies that DEC/Test Manager begin an interactive terminal session taking all input from the specified INPUT file until the INPUT file is exhausted. The /FILTER qualifier can now specify a file version filter and more general date filters.
REVIEW	The REVIEW command now takes the /[NO]READ_ONLY qualifier. The /READ_ONLY qualifier specifies that DEC/Test Manager invoke the Review subsystem in read-only mode. There can be any number of read-only reviewers who can browse through the result descriptions and print files. There can be only one primary reviewer (a reviewer who is not a read-only reviewer) who can issue all Review subsystem commands.
DTM_REVIEW> BACK	The Review subsystem BACK command now takes the /COMPARISON_ABORTED comparison status qualifier.
DTM_REVIEW> INSERT	The Review subsystem INSERT command now takes the /COMPARISON_ABORTED comparison status qualifier and the result-description-expression parameter.
DTM_REVIEW> NEXT	The Review subsystem NEXT command now takes the /COMPARISON_ABORTED comparison status qualifier.
DTM_REVIEW> PCA	When you enter the Review subsystem PCA command to spawn the Analyzer of the VAX Performance and Coverage Analyzer, DEC/Test Manager now takes all the globally defined symbols for your process.

Table 2 (Cont.): Modified Commands

Command	Description
DTM_REVIEW> PRINT	The Review subsystem PRINT command now takes the /COMPARISON_ABORTED comparison status qualifier, the /[NO]LOG qualifier, the /SELECTED qualifier, and the result-description-expression parameter.
DTM_REVIEW> SHOW	The Review subsystem SHOW command now takes the /COMPARISON_ABORTED comparison status qualifier, the /FILES qualifier, and the result-description-expression parameter. The /FILES qualifier specifies that DEC/Test Manager display information about the specified result description. The /SUMMARY qualifier now specifies that DEC/Test Manager display the Collection Summary Information only.
DTM_REVIEW> UPDATE	The Review subsystem UPDATE command now takes the /[NO]LOG qualifier and the result-description-expression parameter.
SHOW GROUP	The SHOW GROUP command now takes the /FULL qualifier. This qualifier specifies the amount of information DEC/Test Manager should display.
STOP	The /NOCONFIRM qualifier is now the default for the STOP command.
SUBMIT	The /CONFIRM qualifier is now the default for the SUBMIT command if the collection has already been run.

Introduction to DEC/Test Manager

This chapter introduces you to DEC/Test Manager, a tool for software development and maintenance. DEC/Test Manager organizes and automates the software regression testing process.

This chapter is divided into three sections:

- A brief overview that describes DEC/Test Manager, regression testing, and how DEC/Test Manager automates the regression testing process
- An example showing how to use the fundamental DEC/Test Manager functions and introducing DEC/Test Manager terminology
- A summary of the DEC/Test Manager commands grouped by function

1.1 Product Overview

DEC/Test Manager is a software development and maintenance tool that provides an efficient, automated way to run, review, and store software regression tests and test results.

DEC/Test Manager is based on the concept of **regression testing**. Regression testing is a method of ensuring that a program being developed runs correctly and that new features added to a program do not affect the correct execution of previously tested features.

In regression testing, you run established software tests and compare the actual test results with the expected results. If the actual results do not agree with the expected results, the software being tested may contain errors. If errors do exist, the software being tested is said to have *regressed*.

1.1.1 Steps in Regression Testing Without DEC/Test Manager

The following list is the typical sequence of steps to follow when performing regression testing without using DEC/Test Manager:

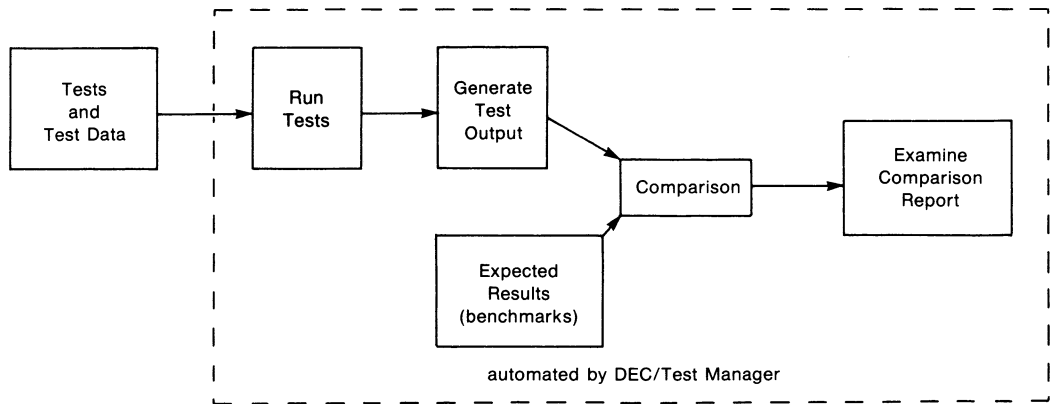
1. Create tests for your software.
2. Organize your tests and create a mechanism to allow ready access to tests as needed.
3. Run your tests.
4. Examine the test results.
 - Compare the actual test results to those you expected and note any differences.
 - For incorrect test output, revise the program code to correct the problem. Repeat steps 3 and 4 until the test output is correct.
 - Save the correct output as the validated test results.
5. Repeat steps 3 and 4 whenever you modify the program or add new code.
 - If the current and the previously validated test results match, the program being tested is working as expected.
 - If you find unexpected changes in test results, the program being tested may contain errors.

Correct the program you are testing and rerun the tests whose results did not match. Repeat this cycle until all results are valid. For future test runs, use these validated test results as references against which to compare the current test results.

DEC/Test Manager automates steps 2 through 4. You still must create your own tests and test data. DEC/Test Manager automates the remainder of the testing procedure.

Figure 1-1 shows the sequence of steps in regression testing. The outlined area indicates those steps DEC/Test Manager automates.

Figure 1-1: DEC/Test Manager and Regression Testing



ZK-2084-84

1.1.2 Steps in Regression Testing with DEC/Test Manager

The following list is the typical sequence of steps to follow when performing regression testing using DEC/Test Manager:

1. Set up a DEC/Test Manager system.
 - Create a DEC/Test Manager library.
 - Create tests by writing command files to test your software or by recording interactive terminal sessions.
 - Identify each test and its related files to DEC/Test Manager.
 - Categorize your tests, if desired, by placing them in groups.
2. Run your tests.
 - Use DEC/Test Manager to collect the test or set of tests you want to run.
 - Run the collection of tests interactively or in batch.

3. Compare the current test results with the expected results for each test. DEC/Test Manager automatically compares test results with the expected results for each test and records any differences in a difference file.
4. Examine the test results. DEC/Test Manager provides an interactive subsystem that lets you access test results immediately. For easy retesting, DEC/Test Manager also gives you the ability to update or create benchmark files and to group all tests that produce incorrect results.
5. Repeat steps 3 through 5 whenever you modify the program or add new code.

1.1.3 Getting HELP

Online HELP information is available for DEC/Test Manager. Type the following command at the DCL prompt for help with DEC/Test Manager:

```
⌘ HELP DTM
```

To get help with a specific DEC/Test Manager command, for example, the DTM COPY TEST_DESCRIPTION command, type the following:

```
⌘ HELP DTM COPY TEST_DESCRIPTION
```

To get help with the DTM COPY TEST_DESCRIPTION command when you are using DEC/Test Manager as a subsystem, type the following command at the DEC/Test Manager prompt:

```
DTM> HELP COPY TEST_DESCRIPTION
```

Online HELP with the DEC/Test Manager Review commands is available from within the Review subsystem. For example, to get help with the SHOW/SUMMARY command, type the following:

```
DTM_REVIEW> HELP SHOW/SUMMARY
```

1.2 Getting Started

This section introduces the basic features of DEC/Test Manager. It is intended to get you started using DEC/Test Manager by showing the most fundamental DEC/Test Manager features and terminology. It does not describe advanced features, such as those used to organize a large or complicated test set; this information is covered in later chapters.

The examples presented in this section are designed so that you can recreate this DEC/Test Manager session on your screen. Typing in all the examples should take you about an hour. Working with these examples will show you how to do the following:

- Invoke DEC/Test Manager.
- Create a new DEC/Test Manager library.
- Select an existing DEC/Test Manager library.
- Create a noninteractive test and identify it to DEC/Test Manager.
- Create an interactive test and identify it to DEC/Test Manager.
- Organize your tests so they can be run together.
- Run your tests.
- Examine your test results and create a benchmark file.

1.2.1 Invoking DEC/Test Manager

You can use DEC/Test Manager in two ways:

- Invoke DEC/Test Manager as a subsystem by entering DTM at the DCL prompt. DEC/Test Manager displays the subsystem prompt DTM> .

```
⌘ DTM
DTM>
```

The DEC/Test Manager commands you enter should not be prefaced with DTM. For example:

```
DTM> SHOW VERSION
DEC/Test Manager Version V2.1
DTM>
```

Each time you enter a command, the command is executed, control is returned to the DEC/Test Manager subsystem command level, and the DEC/Test Manager prompt is displayed. To terminate the DEC/Test Manager session, type EXIT or press CTRL/Z. Control is returned to DCL command level.

```
DTM> EXIT
$
```

- Enter DEC/Test Manager commands at the DIGITAL Command Language (DCL) dollar sign (\$) prompt by prefacing each command with DTM. After each command executes, control is returned to the DCL command level.

```
$ DTM SHOW VERSION
DEC/Test Manager Version V2.1
$
```

If you plan to enter a number of commands, you should use DEC/Test Manager as a subsystem to avoid the overhead involved with invoking DEC/Test Manager multiple times. However, for demonstration purposes, the examples in this manual show commands entered at the DCL prompt.

1.2.2 Creating and Selecting a Library

DEC/Test Manager stores all the information it needs to manage a test system in an area called a DEC/Test Manager **library**. The library is a specially formatted VAX/VMS directory. After you create a DEC/Test Manager library, the directory containing that library should be reserved exclusively for use by DEC/Test Manager.

Before you use DEC/Test Manager, you must create a DEC/Test Manager library in an empty VAX/VMS directory. Thereafter, whenever you use DEC/Test Manager, point DEC/Test Manager to the library you want to use.

1.2.2.1 Creating a Library—CREATE LIBRARY

There are two steps in creating a DEC/Test Manager library:

1. Create a VAX/VMS directory by typing

```
⌘ CREATE/DIRECTORY [.DTMLIB]
⌘
```

This creates a new, empty subdirectory of your current default directory.

2. Create a library in that directory by typing

```
⌘ DTM CREATE LIBRARY [.DTMLIB] " New DEC/Test Manager library"
⌘DTM-S-CREATED, DEC/Test Manager library DRA1:[PROJECT.DTMLIB] created
⌘
```

The phrase enclosed in quotes following the library specification is a **remark**. It enables you to associate a comment with the library you are creating. DEC/Test Manager prompts you for a remark if you do not include one on the command line.

NOTE

Do not create subdirectories or files in a directory containing the DEC/Test Manager library or any of its subdirectories. Do not set your default directory to be a DEC/Test Manager library or any of its subdirectories.

1.2.2.2 Selecting an Existing Library—SET LIBRARY

For subsequent sessions, after you have created one or more DEC/Test Manager libraries, point DEC/Test Manager to the library you want to use. For example, to select the library you created in the previous example, type

```
⌘ DTM SET LIBRARY [.DTMLIB]
⌘DTM-S-LIBIS, DEC/Test Manager library is DRA1:[PROJECT.DTMLIB]
⌘
```

After you have selected a library, all DEC/Test Manager commands you enter refer to that library until you select another library, create another library, or log out.

The CREATE LIBRARY command performs an implicit SET LIBRARY command so that you can now use DEC/Test Manager commands with this library.

1.2.3 Creating Tests and Setting Up a Test System

After you have created a DEC/Test Manager library, you can begin to create and organize a test system. To do this, create your tests, then identify the tests and their associated files to DEC/Test Manager.

In addition to the library, DEC/Test Manager provides a number of structures for organizing your test system. The most fundamental of these structures is the **test description**. A test description identifies a test to DEC/Test Manager by its **test-name**. A test description consists of a series of fields whose contents point to files and other information needed to run the test.

The core of each test description is the **template file**. Every test description must have a template file. For noninteractive tests, the template file is a DCL command procedure or it is the test itself. For interactive tests, the template file is the **SESSION file** created with the CREATE TEST_DESCRIPTION/RECORD or MODIFY TEST_DESCRIPTION/RECORD command.

You can optionally specify a **test prologue** file and a **test epilogue** file. These are command procedures that extend the test environment. The test prologue file is run immediately before the template file and can function as a setup file. The test epilogue file is run immediately after the template file and can function as a cleanup file or as a filter for test results.

The **benchmark file** contains the expected output for the test's execution.

The following subsections show you how to create noninteractive and interactive tests.

1.2.3.1 Creating a Noninteractive Test

DEC/Test Manager does not provide an automated means of creating **noninteractive tests**—tests whose template files are command files. You must invoke a text editor to create the test and the test template file. Then use the CREATE TEST_DESCRIPTION command to create a test description for the test.

Writing the Test

In the following example, you are testing DIGITAL Standard Runoff (DSR) and you want to test the left margin command. Invoke a text editor (EDT is used in these examples), change to screen mode editing, and create the following file, LEFTMARGIN1.RNO, a simple test of the DSR left margin command. Then exit the editor and save LEFTMARGIN1.RNO.

```
⌘ EDIT/EDT LEFTMARGIN1.RNO
Input file does not exist
[EOB]
* CHANGE
This line should appear at the default left margin (0)
.lm15
This line should appear 15 spaces from the default left margin
.lm+10
This line should appear 25 spaces from the default left margin
.lm-20
This line should appear 5 spaces from the default left margin
~Z
[EOB]
* EXIT
```

Creating the Template File

Next, create the template file LMTEMPLATE.COM to run your test LEFTMARGIN1.RNO. The template file LMTEMPLATE.COM is a simple VAX/VMS command file that runs DSR over your test. Invoke a text editor and create LMTEMPLATE.COM. Be sure to substitute your default directory specification for the directory specification shown in the example (DRA1:[PROJECT.RNO]). To make the test file and the resulting MEM output file part of the test run record, type both files to SYS\$OUTPUT, as shown in the following example.

```

$ EDIT/EDT LMTEMPLATE.COM
Input file does not exist
[EOB]
* CHANGE
$ ! TEMPLATE for a simple test of the DSR left margin (.LM) command
$ !
$ ! Display the input file
$ !
$ TYPE DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ ! Run DSR on the left margin test
$ !
$ DSR DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ ! Show the output of this runoff test
$ !
$ TYPE LEFTMARGIN1.MEM
$ !
$ ! Cleanup
$ !
$ DELETE LEFTMARGIN1.MEM;1
~Z
[EOB]
* EXIT

```

Creating a Test Description—CREATE TEST_DESCRIPTION

Now identify the test and its template file to DEC/Test Manager by using the DTM CREATE TEST_DESCRIPTION command to create a test description with test-name LMTEST1. Test description LMTEST1 references your template file, LMTEMPLATE.COM, thus identifying to DEC/Test Manager both the template file and the test it runs. Be sure to substitute your default directory specification for the directory specification shown in this example (DRA1:[PROJECT.COM]).

To create test description LMTEST1, type the following continued command line:

```

$ DTM CREATE TEST_DESCRIPTION lmtest1 "simple test of left margin command" -
_ $ /TEMPLATE=dra1:[project.com]lmtemplate.com
%DTM-I-DEFAULTED, benchmark file name defaulted to LMTEST1.BMK
%DTM-S-CREATED, test description LMTEST1 created
$

```

The phrase enclosed in quotes following the test-name is a **remark**. It enables you to associate a comment with the test description you are creating. DEC/Test Manager prompts you for a remark if you do not include one on the command line.

1.2.3.2 Creating an Interactive Test—CREATE TEST_DESCRIPTION/RECORD

DEC/Test Manager provides a means of assisting you when you create **interactive tests**—tests whose template files are **SESSION files**.

You must enter the CREATE TEST_DESCRIPTION/RECORD command to record an interactive terminal session in a SESSION file. After recording the terminal session, DEC/Test Manager creates a test description for the test, makes the SESSION file the template file for the test, and associates any other files with the test.

In the following example, you need to create a template that contains the DSR commands and standard text you always use in writing your memos. Invoke a text editor and create MEMO_TEMPLATE.RNO by typing the template for writing memos shown here.

```
EDIT/EDT MEMO_TEMPLATE.RNO
Input file does not exist
[EOB]
* CHANGE
.PS60,70
.FLAGS BOLD
.FLAGS SUBSTITUTE
.B
.NF
.TS28
      ^*I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M^*
.TS39,53.LM39
.B2
^*To:\*
^*Date:\*
^*From:\*
^*Ext:\*
.TS20.LMO.B3
^*Subject:\*
.B2
.F.J

. ! Input the text of your memo starting on the next line.
^Z
* EXIT
```

You are now ready to test your template for writing memos.

Creating the Template File and the Test Description

Type the CREATE TEST_DESCRIPTION command with the /RECORD qualifier to record an interactive terminal session while you create a memo using the memo template. If you do not include the /TEMPLATE qualifier to specify a file-name for the test template file, DEC/Test Manager automatically creates the test template file (MEMO_TEST.SESSION) for your interactive test.

```

$ DTM CREATE TEST_DESCRIPTION/RECORD memo_test -
_ $ "first interactive test of memo template"
%DTM-I-DEFAULTED, benchmark file name defaulted to MEMO_TEST.BMK
%DTM-I-DEFAULTED, template file name defaulted to MEMO_TEST.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.
```

\$

Everything you type and all output from the system is recorded in the MEMO_TEST.SESSION file (the test template file).

Invoke a text editor (EDT is used in these examples) and create the DSR memo file MEMO.RNO.

```

$ EDIT/EDT memo.rno
Input file does not exist
[EOB]
*
```

Include the memo template in the editing session and change to screen mode editing.

```

.
.
.
* INCLUDE memo_template.rno
* CHANGE
```

The memo template displays on the screen.

```
.PS60,70
.FLAGS BOLD
.FLAGS SUBSTITUTE
.B
.NF
.TS28
      ^*I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M\*
.TS39,53.LM39
.B2
^*To:\*
^*Date:\*
^*From:\*
^*Ext:\*
.TS20.LMO.B3
^*Subject:\*
.B2
.F.J
```

.! Input the text of your memo starting on the next line.

[EOB]

Enter the text of the memo. Everything you type is recorded in the test template file MEMO_TEST.SESSION.

```
PS60,70
.FLAGS BOLD
.FLAGS SUBSTITUTE
.B
.NF
.TS28
      ^*I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M\*
.TS39,53.LM39
.B2
^*To:\* John Dow
^*Date:\* Jan 8, 1986
^*From:\* Sally Smith
^*Ext:\* 1234
.TS20.LMO.B3
^*Subject:\* The new memo template.
.B2
.F.J
```

.! Input the text of your memo starting on the next line.

This is the new memo template for creating memos using RUNOFF. Let me know how you like using it.

[EOB]

Exit the editor and save MEMO.RNO.

```
.  
.  
.  
RUNOFF. Let me know how you like using it.  
^Z  
[EOB]
```

```
*EXIT  
DRA1: [PROJECT]MEMO.RNO;1 25 LINES
```

```
§
```

Enter commands first to create a MEM file, then to type the MEM file, and finally to delete both the RNO and MEM files.

```
§ RUNOFF MEMO.RNO  
§ TYPE MEMO.MEM
```

I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M

To: John Dow
Date: Jan 8, 1986
From: Sally Smith
Ext: 1234

Subject: The new memo template.

This is the new template for creating memos using RUNOFF. Let me know how you like using it.

```
§ DELETE MEMO.RNO;1  
§ DELETE MEMO.MEM;1  
§
```

Terminate the interactive terminal session by entering the **termination character** CTRL/P twice. Note that only one CTRL/P echoes on the screen.

```
§
```

```
^P
```

```
%DTM-I-BMK_SAVED, benchmark has been saved in file  
DRA1: [PROJECT.DTMLIB] MEMO_TEST.BMK;1  
%DTM-S-RECORDED, test MEMO_TEST has been successfully recorded in file  
DRA1: [PROJECT]MEMO_TEST.SESSION  
%DTM-S-CREATED, test description MEMO_TEST created  
§
```

1.2.4 Running Tests

Running tests with DEC/Test Manager is a two-step procedure. First, you create a **collection** of tests and then you choose whether to submit your collection of tests for batch processing or to run your collection of tests interactively.

Every collection is identified with a unique **collection-name**, which identifies the tests in the collection as a set. You can optionally supply a **collection prologue** file and a **collection epilogue** file for a collection. The collection prologue file is a command file that can function as a setup file. It runs at the beginning of the collection, before any tests are run. The collection epilogue file is a command file that can function as a cleanup file. It runs at the end of the collection, after the tests have run and the test results have been compared.

1.2.4.1 Creating Collections—CREATE COLLECTION

In the following example, you want to run both your simple left margin test (LMTEST1) and your memo test (MEMO_TEST) together and give them the collection-name RNOCOLLECT. Create a collection containing both tests by typing the following command line:

```
$ DTM CREATE COLLECTION  rnocollect lntest1,memo_test -  
_ $ "left margin test and memo test"  
%DTM-S-CREATED, collection rnocollect created  
$
```

1.2.4.2 Executing Collections

After you create your collection, you must decide whether to submit your collection for batch processing or to run your collection interactively. DEC/Test Manager can execute collections of tests either interactively or in batch. You can execute both interactive and noninteractive tests in either mode without affecting the test results, provided your batch and interactive environments are the same.

Either way you execute your tests, DEC/Test Manager creates a **result file** and places the results of the test run in this file.

DEC/Test Manager then compares the result and benchmark files for the test (if the benchmark file exists) and saves the result file in the DEC/Test Manager library. Benchmark files contain expected test output and are created and updated while reviewing the test results. DEC/Test Manager stores the status of the comparison (for example, SUCCESSFUL,

UNSUCCESSFUL, or NEW TEST) along with any differences in a **difference file** in the DEC/Test Manager library.

When you interactively run a collection, by default DEC/Test Manager displays the contents of the result file on your screen as it writes the file and then performs an automatic comparison.

The next two sections show how to use DEC/Test Manager to execute RNOCOLLECT, first by submitting the collection to the batch queue and then by running it interactively. Normally, you would execute a collection only once before reviewing it.

Submitting a Collection to Batch—SUBMIT

The DEC/Test Manager SUBMIT command executes a collection of tests as a batch job. To submit RNOCOLLECT to the batch queue, type

```
$ DTM SUBMIT rnocollect /NOTIFY
%DTM-S-SUBMITTED, collection RNOCOLLECT submitted
-DTM-I-TEXT, Job RNOCOLLECT (queue SYS$BATCH, entry 2222) started on SYS$BATCH
$
```

DEC/Test Manager may take some time to run a collection depending on the complexity and number of tests in the collection. Note that when you include the /NOTIFY qualifier, the system notifies you when the batch job is finished.

```
$
Job RNOCOLLECT (queue SYS$BATCH, entry 2222) completed
$
```

Running a Collection Interactively—RUN

The DEC/Test Manager RUN command executes a collection interactively on your screen. If you submitted RNOCOLLECT for batch processing, you must wait for it to finish before continuing. To execute collection RNOCOLLECT interactively, type

```
$ DTM RUN rnocollect
Collection RNOCOLLECT has not been reviewed. Confirm rerun [Y/N] (N):
```

DEC/Test Manager prompts you to specify whether you want to execute the collection a second time without reviewing it first. Type Y to run the collection.

Collection RNOCOLLECT has not been reviewed. Confirm rerun [Y/N] (N): Y

Starting LMTEST1 test run...

This line should appear at the default left margin (0)

.lm15

This line should appear 15 spaces from the default left margin

.lm+10

This line should appear 25 spaces from the default left margin

.lm-20

This line should appear 5 spaces from the default left margin

This line should appear at the default left margin (0)

This line should appear 15 spaces from the default left margin

This line should appear 25 spaces from the default left margin

This line should appear 5 spaces from the default left margin

Starting MEMO_TEST test run...

\$

\$ EDIT/EDT memo.rno

Input file does not exist

[EOB]

* INCLUDE memo_template.rno

* CHANGE

.

.

The blank template for memos appears on the screen and everything that you entered when you recorded the interactive terminal session in MEMO_TEST.SESSION is replayed on the screen. You see the commands that exit the editor and process the file with RUNOFF. Then you see the processed MEM file on the screen.

You see the commands that delete MEMO.RNO and MEMO.MEM, and a message stating that DEC/Test Manager is cleaning up the library and comparing the test results with the benchmark file. The DCL prompt reappears when the collection run is finished.

\$ DELETE MEMO.RNO;1

\$ DELETE MEMO.MEM;1

\$

Performing post-run cleanup with comparison...

\$

1.2.5 Examining Test Results—REVIEW

You can examine test result files with the DEC/Test Manager Review subsystem. The Review subsystem allows you to explore interactively the results of each test, to print files related to the test results, and to create or update the test's benchmark file.

DEC/Test Manager generates a **result description** for each test in a collection. A result description identifies the output files for each test and indicates the status of each test. You can reference the information in the result description for a test by its **result-description-name**, which is the same as the corresponding test-name.

To review your test results when the collection is finished running, type

```
$ DTM REVIEW rnocollect
Collection RNOCOLLECT with 2 tests was created on 08-JAN-1986 07:19:16 by the
command:
  CREATE COLLECTION rnocollect lmtest1,memo_test left margin test and memo_test
  Last Review Status = not previously reviewed
  Success count = 1
  Unsuccessful count = 0
  New test count = 1
  Updated test count = 0
  Comparisons aborted = 0
  Test not run count = 0
DTM_REVIEW>
```

To review the results of test LMTEST1, enter NEXT to display the first result description.

```
DTM_REVIEW> NEXT
Result Description LMTEST1          Comparison status : New test
```

In this case, the result description for LMTEST1 tells you that LMTEST1 is a new test; therefore, no benchmark or difference file exists. As your next step, you may want to examine the result file for LMTEST1 to determine whether the results are correct. To do this, type

```
DTM_REVIEW> SHOW/RESULT
Result file DRA1:[PROJECT.DTMLIB.RNOCOLLECT]LMTEST1.RES For Result
Description LMTEST1
$ ! TEMPLATE for a simple test of the DSR left margin (.LM) command
$ !
$ ! Display the input file
$ !
$ TYPE DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
```

```

This line should appear at the default left margin (0)
.lm15
This line should appear 15 spaces from the default left margin
.lm+10
This line should appear 25 spaces from the default left margin
.lm-20
This line should appear 5 spaces from the default left margin
$ !
$ ! Run DSR on the left margin test
$ !
$ DSR DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ ! Show the output of this runoff test
$ !
$ TYPE LEFTMARGIN1.MEM

```

This line should appear at the default left margin (0)

This line should appear 15 spaces from the default left margin

This line should appear 25 spaces from the default left margin

This line should appear 5 spaces from the default left margin

```

$ !
$ ! Cleanup
$ !
$ DELETE LEFTMARGIN1.MEM;
DTM_REVIEW>

```

The result file contains the expected results. You can make this file the benchmark file for LMTEST1 by typing

```

DTM_REVIEW> UPDATE
%DTM-I-UPDATED, the benchmark for test LMTEST1 has been updated
DTM_REVIEW>

```

To print the newly created benchmark file, type

```

DTM_REVIEW> PRINT/BENCHMARK
%DTM-S-PRINT, file DRA1:[PROJECT.DTMLIB]LMTEST1.BMK;1 of test LMTEST1
selected for printing
DTM_REVIEW>

```

Note that the SHOW, UPDATE, and PRINT commands all refer to the currently selected result description. The benchmark file is updated immediately after you enter the PRINT command. Files marked for printing are printed when you exit the Review subsystem.

To review the results for test MEMO_TEST, enter the SELECT command and specify the result-description-name MEMO_TEST to display its result description.

```
DTM_REVIEW> SELECT MEMO_TEST
Result Description MEMO_TEST      Comparison Status: Successful
DTM_REVIEW>
```

Because a benchmark file exists for MEMO_TEST and the results of the test run as contained in the result file match the benchmark file, MEMO_TEST was successful.

1.2.6 Exiting the Review Subsystem—EXIT

To exit the Review subsystem and print the file you selected for printing, type

```
DTM_REVIEW> EXIT
%DTM-S-PRINTQD, print job has been sent to the print queue
-DTM-I-TEXT, Job LMTEST1 (queue SYS$PRINT, entry 1666) started on LPA0
%DTM-S-EXIT, leaving Review subsystem
$
```

Pressing CTRL/Z will also exit the Review subsystem and print selected files.

To exit the Review subsystem without printing the files you marked, type

```
DTM_REVIEW> EXIT/NOPRINT
%DTM-S-EXIT, leaving Review subsystem
$
```

Pressing CTRL/C will also exit the Review subsystem without printing the selected file.

1.3 Command Summaries

This section lists the DEC/Test Manager commands grouped by function. The Command Dictionary contains a complete description of each command.

1.3.1 Invoking DEC/Test Manager

The commands described in Table 1-1 get you started using DEC/Test Manager.

Table 1-1: Commands to Get Started Using DEC/Test Manager

Command	Description
DTM	Invokes DEC/Test Manager as a subsystem
DTM/NOINIT	Invokes DEC/Test Manager as a subsystem without executing a DEC/Test Manager initialization file
EXIT (CTRL/Z)	Terminates the DEC/Test Manager subsystem and returns control to DCL
HELP	Displays help information about DEC/Test Manager
SHOW VERSION	Displays the version of DEC/Test Manager you are using

1.3.2 Creating or Selecting a DEC/Test Manager Library

The commands described in Table 1-2 allow you to create or select a DEC/Test Manager library.

Table 1-2: Commands for Creating or Selecting a DEC/Test Manager Library

Command	Description
CREATE LIBRARY	Creates a DEC/Test Manager library in an empty directory
CONVERT LIBRARY	Converts an existing DEC/Test Manager library for use with the current version of DEC/Test Manager
SET LIBRARY	Selects an existing DEC/Test Manager library as the current library
SHOW LIBRARY	Displays the directory specification for the current DEC/Test Manager library

1.3.3 Using a DEC/Test Manager Library

The commands described in Table 1-3 are general function DEC/Test Manager commands. Some of these commands alter the current DEC/Test Manager library or display information about it. Others allow you to execute DEC/Test Manager test sessions more easily by managing the current test environment.

Table 1-3: General Function DEC/Test Manager Commands

Command	Description
ATTACH	Connects your terminal to another process in your job.
@file-spec	Executes the specified command file.
DEFINE/KEY	Defines or redefines keypad keys and control sequences.
REVIEW	Invokes the DEC/Test Manager Review subsystem to review test results.
REVIEW/READ_ONLY	Invokes the DEC/Test Manager Review subsystem in read-only mode to browse through test results and print files. You cannot enter the INSERT or UPDATE commands.
SPAWN	Creates a subprocess of your current process.
VERIFY	Checks the structure of the current DEC/Test Manager library and the structure of the files in that library.
VERIFY/RECOVER	Corrects some errors that the VERIFY command reports and restores the library after a system crash.

1.3.4 Logging History Information

The commands described in Table 1-4 are DEC/Test Manager commands used to manage the history information logged by DEC/Test Manager.

Table 1-4: History Logging Commands

Command	Description
DELETE HISTORY	Deletes information from the DEC/Test Manager history file
REMARK	Inserts a comment into the DEC/Test Manager history file
SHOW HISTORY	Displays information from the DEC/Test Manager history file

1.3.5 Creating and Maintaining Tests

The commands described in Table 1-5 allow you to create and maintain tests in the current DEC/Test Manager library.

Table 1-5: Commands for Creating and Maintaining Tests

Command	Description
COPY TEST_DESCRIPTION	Creates an exact or modified copy of an existing test description
CREATE TEST_DESCRIPTION	Creates a test description for a noninteractive test
CREATE TEST_DESCRIPTION/RECORD	Records an interactive terminal session and creates a test description for it
CREATE TEST_DESCRIPTION/INPUT	Generates a SESSION file from an INPUT file
DELETE TEST_DESCRIPTION	Deletes test descriptions
DISPLAY/BENCHMARK	Displays the benchmark file for an interactive test
EXTRACT	Generates an INPUT file for editing from a SESSION file
MODIFY TEST_DESCRIPTION	Modifies existing test descriptions
PLAY	Executes an interactive test session
SHOW TEST_DESCRIPTION	Displays the contents of existing test descriptions

1.3.6 Categorizing Tests with Groups

The commands described in Table 1–6 allow you to categorize tests by placing them in groups. Groups of tests can be manipulated as a unit.

Table 1–6: Commands for Categorizing Tests with Groups

Command	Description
CREATE GROUP	Creates an empty group
DELETE GROUP	Deletes empty groups
INSERT GROUP	Includes groups in other groups
INSERT TEST_DESCRIPTION	Includes test descriptions in a group
MODIFY GROUP	Removes or replaces the remark associated with an existing group
REMOVE GROUP	Removes groups from other groups
REMOVE TEST_DESCRIPTION	Removes test descriptions from groups
SHOW GROUP	Displays groups and their contents

1.3.7 Generalizing Test-Related Files with Variables

The commands described in Table 1–7 allow you to use variables to generalize the template, prologue, and epilogue files associated with your tests. By using variables in these files, the files can then be used with multiple tests.

Table 1–7: Commands for Generalizing Test-Related Files with Variables

Command	Description
CREATE VARIABLE	Creates a variable
DELETE VARIABLE	Deletes variables
MODIFY VARIABLE	Modifies variables
SHOW VARIABLE	Displays information about variables

1.3.8 Organizing Test-Related Files

The commands described in Table 1-8 allow you to organize the template, benchmark, prologue, and epilogue files associated with your tests.

Table 1-8: Commands for Organizing Test-Related Files

Command	Description
SET BENCHMARK_DIRECTORY	Establishes a default directory for benchmark files
SET TEMPLATE_DIRECTORY	Establishes a default directory for template files
SET PROLOGUE	Establishes a default collection prologue file
SET EPILOGUE	Establishes a default collection epilogue file
SHOW ALL	Displays the current default directory specifications for benchmark and template directories, and the file specifications for the current collection prologue and collection epilogue files and the number of collections, groups, and tests in the library
SHOW BENCHMARK_DIRECTORY	Displays the current default directory specification for benchmark files
SHOW TEMPLATE_DIRECTORY	Displays the current default directory specification for template files
SHOW PROLOGUE	Displays the file specification for the current default collection prologue file
SHOW EPILOGUE	Displays the file specification for the current default collection epilogue file

1.3.9 Executing Tests

The commands described in Table 1-9 allow you to execute tests.

Table 1-9: Commands for Executing Tests

Command	Description
CREATE COLLECTION	Creates a collection of tests
COMPARE	Manually compares the result and benchmark files for tests in a collection
DELETE COLLECTION	Deletes collections
RECREATE	Recreates an existing collection by first deleting the collection and certain files related to it and then creating the collection again using the current information in your DEC/Test Manager library
RUN	Executes a collection interactively
SHOW COLLECTION	Displays information about collections
STOP	Stops execution of a collection that has been submitted to the batch queue
SUBMIT	Executes a collection in batch

1.3.10 Reviewing Test Results

The DTM REVIEW command invokes the DEC/Test Manager Review subsystem. While in the Review subsystem, you can issue subsystem commands to review interactively the result description for each test in a specified collection.

The DTM REVIEW/READ_ONLY command invokes the DEC/Test Manager Review subsystem in read-only mode. While in read-only mode, you can browse through the test results, display information, and print files. You cannot issue subsystem commands that make changes to the result descriptions such as INSERT and UPDATE.

1.3.10.1 Using the Review Subsystem

The commands described in Table 1-10 allow you to establish the Review subsystem environment and to perform general functions.

Table 1-10: General Review Subsystem Commands

Command	Description
ATTACH	Connects your terminal to another process in your job
@file-spec	Executes the specified DEC/Test Manager command file
DEFINE/KEY	Defines or redefines keys on the Review subsystem keypad
EXIT (CTRL/Z)	Terminates the Review session, prints marked files, and inserts marked test descriptions into the group
EXIT/NOPRINT/NOINSERT (CTRL/C)	Terminates the Review session without printing files or inserting test descriptions into the group
HELP	Provides information on the Review subsystem commands
PCA	Invokes the Analyzer of the VAX Performance and Coverage Analyzer
SPAWN	Creates a subprocess of your current process

1.3.10.2 Selecting Test Results to Review

The commands described in Table 1-11 allow you to select specific test result descriptions to review.

Table 1-11: Commands to Select Test Results to Review

Command	Description
BACK	Moves backward through the result descriptions
FIRST	Selects the first result description
LAST	Selects the last result description
NEXT	Moves forward through the result descriptions
SELECT	Selects the specified result description

Note that pressing the RETURN key performs the same function as the NEXT command.

1.3.10.3 Manipulating Test Results

The commands described in Table 1-12 allow you to manipulate the test results you are reviewing.

Table 1-12: Commands to Manipulate Result Descriptions

Command	Description
INSERT	Inserts specified test descriptions into a group
PRINT	Prints specified files
SHOW	Displays output files and information about result descriptions
SHOW/FILES	Displays information about output files
SHOW/SUMMARY	Displays the Collection Summary Information
UPDATE	Updates or creates benchmark files

Getting Started with DEC/Test Manager

This chapter describes how to set up a DEC/Test Manager library. It includes information on the following:

- Organizing a DEC/Test Manager library
- Defining access to a DEC/Test Manager library
- Creating a new DEC/Test Manager library
- Converting a DEC/Test Manager library created with an earlier version of DEC/Test Manager
- Selecting an existing DEC/Test Manager library

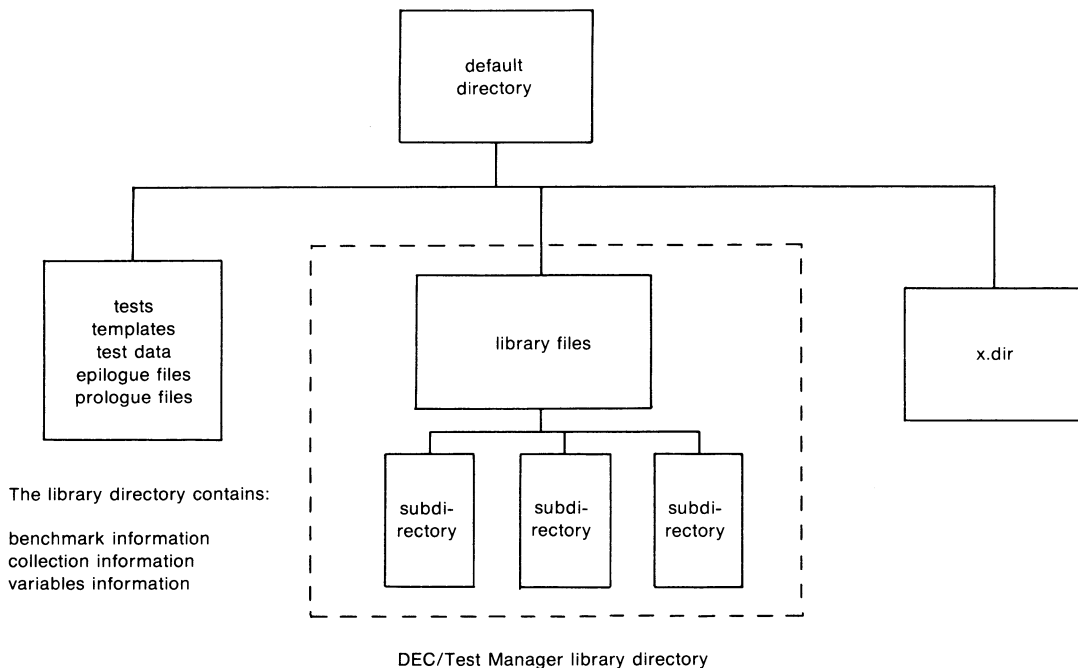
To begin using DEC/Test Manager, you must establish a library, create tests, and describe these tests to DEC/Test Manager. You can use any of the DEC/Test Manager optional features to enhance your test system, although none of them are required to use DEC/Test Manager. For more information about optional features, see Chapter 6.

2.1 Defining a DEC/Test Manager Library

A DEC/Test Manager **library** is a VAX/VMS directory that is used to store result files, benchmark files, and the files that DEC/Test Manager uses to manage the test system. You can optionally store benchmark files outside the DEC/Test Manager library in another directory or in a DEC/CMS library. You should use only DEC/Test Manager commands to modify files in a DEC/Test Manager library.

Figure 2-1 shows a directory structure for a DEC/Test Manager test system.

Figure 2-1: The DEC/Test Manager Test System



ZK-2082-84

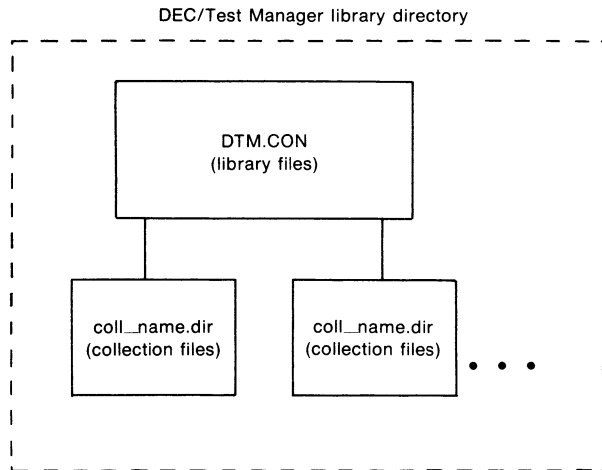
NOTE

Never set your default directory to a DEC/Test Manager library or any of its subdirectories.

Do not create or modify files in a library directory. User-created or user-modified files could interfere with test processing when you run collections of tests. DEC/Test Manager can also delete any files it does not recognize.

Each project typically uses a single library, although you can create more than one library for any project. Using one DEC/Test Manager library for each project can make it easier to manage and to access all tests within the project. You may also want to create a personal library where you can develop new tests. Figure 2-2 shows a DEC/Test Manager library directory.

Figure 2-2: DEC/Test Manager Library Directory



ZK-2083-84

Store template files and tests that you create, as well as all prologue and epilogue files, outside the DEC/Test Manager library. These files can be created and modified with standard DCL commands. For convenience, you may want to store these files in one directory and make your DEC/Test Manager library a subdirectory of this directory. You can also store templates, benchmarks, prologues, and epilogues in one or more DEC/CMS libraries.

2.2 Defining Access to a Library

DEC/Test Manager does not define protection for library directories or library files. To use all DEC/Test Manager commands in a particular library, you must have the following access:

- Read and write access to the DEC/Test Manager library directory
- Read and write access to the library control file (00DTM.CON) and to the history file (00DTM.HIS)
- Read and write access to all collection subdirectories (collection-name.DIR) in the library
- Write access to all collection control files (collection-name.CON) in the collection subdirectories
- Read and delete access to the generic command file used to run collections (DTM\$\$TEST_RUN.COM)
- Read and delete access to all other files in the library directory and in the collection subdirectories (collection files and test description files, for example)
- Execute access to the DEC/Test Manager image (SYS\$SYSTEM:DTM.EXE)

Note that if you allow read-only access to a library directory or to the library control file, users cannot make changes to the contents of the library, execute tests, review tests, or perform comparisons. If you do not allow write access to the collection control files, users will not be able to review collections. You should allow group read and write access to these files.

When you create the library, give read, write, and delete access to every file in the library to at least one user. All three types of access are needed to execute the VERIFY/RECOVER command.

NOTE

If you have the SYSPRV privilege, file protection problems may occur when you issue a DEC/Test Manager command that creates files in a directory or library owned by another user. This occurs because SYSPRV changes the ownership of files created in a directory owned by another user.

The User Identification Code (UIC) protection or Access Control Lists (ACLs) determine the access to library files. If you do not use ACLs, all users of a particular DEC/Test Manager library must be in the same user group. If you want to define more selective protection (where various individuals in the user group have differing access), you can use ACLs for the library directory and its files and subdirectories.

The following subsections summarize the procedures you can use to define the access to a DEC/Test Manager library. For more information, see the *VAX/VMS DCL Dictionary* and the *Guide to VAX/VMS System Security*.

2.2.1 UIC Protection

Every file has a UIC protection mask that defines the type of access (read, write, execute, and delete) allowed to various categories of users (system, owner, group, and world). You can use the DCL SET PROTECTION command to specify a particular protection mask for the library directory and for other DEC/Test Manager files. The following example shows a protection mask that allows system, owner, and group access to a library but denies world access to the library contained in the [PROJ].TESTING] directory:

```
⚡ SET PROTECTION=(S:RWE,O:RWE,G:RWE,W) [PROJ]TESTING.DIR
```

2.2.2 ACL Protection

ACLs allow you to grant or deny access to a specified user or group of users on the system. Each access control entry (ACE) in the ACL for a file identifies one or more users and defines the type of access allowed.

Directory ACLs control access to a DEC/Test Manager library in the following three ways:

- Specifying a UIC protection mask to be assigned to each new file created in the directory
- Specifying an identifier-based protection mask to be assigned to each new file created in the directory
- Controlling access to the directory file itself

To propagate a UIC protection mask, use the DEFAULT_PROTECTION keyword in the first field of a directory ACE. For example:

(IDENTIFIER=TESTGRP,OPTIONS=DEFAULT,S:RWED,O:RWED,G:RWED)

This ACE specifies that the UIC protection (S:RWED,O:RWED,G:RWED) is applied to each new file created in the directory. (It does not affect any files that may already exist in the directory.) If no ACEs impose stricter limitations, the system, owner, and group users (as defined by the UIC) are granted full use of the library.

To propagate an identifier-based protection mask, use the OPTIONS=DEFAULT keyword in the second field of a directory ACE. For example:

(IDENTIFIER=TESTGRP,OPTIONS=DEFAULT,ACCESS=READ+WRITE+DELETE)

The OPTIONS=DEFAULT keyword directs the operating system to duplicate this ACE in the ACL of every new file that is created in the library. This ACE then grants read, write, and delete access to users who have the TESTGRP identifier.

Table 2-1 shows a list of the DEC/Test Manager commands and the protection required for each object that the command accesses.

Table 2-1: Access Required for Individual DEC/Test Manager Commands

Command	Library Directory	Library Control File	Collection Subdirectories	Library Files	Collection Files
COMPARE	RW	RW	RW	R	RWD
COPY					
TEST_DESCRIPTION	RW	RW		RW	
CREATE					
COLLECTION	RW	RW	RW	R	RWD
GROUP	RW	RW			
LIBRARY	RW ¹	RW			
TEST_DESCRIPTION	RW	RW		RW	
VARIABLE	RW	RW			

¹The directory must be empty.

Table 2-1 (Cont.): Access Required for Individual DEC/Test Manager Commands

Command	Library Directory	Library Control File	Collection Subdirectories	Library Files	Collection Files
DELETE					
COLLECTION	RW	RW	RD	R	RD
GROUP	RW	RW			
HISTORY	RW	R		RWD	
TEST_DESCRIPTION	RW	RW		RD	
VARIABLE	RW	RW			
DISPLAY	RW	RW		R	
EXTRACT				R	
INSERT					
GROUP	RW	RW			
TEST_DESCRIPTION	RW	RW			
MODIFY					
GROUP	RW	RW			
TEST_DESCRIPTION	RW	RW		RWD	
VARIABLE	RW	RW			
RECREATE	RW	RW	RWD	RWD	RWD
REMOVE					
GROUP	RW	RW			
TEST_DESCRIPTION	RW	RW			
REVIEW	RW	RW	R	RWD	RWD
RUN	RW	RW	RW	RW	RWD
SET					
BENCHMARK_DIRECTORY	RW	RW		RW	
EPILOGUE	RW	RW	RW	RW	RW
LIBRARY	R	R			
PROLOGUE	RW	RW	RW	RW	RW
TEMPLATE_DIRECTORY	RW	RW		RW	

Table 2-1 (Cont.): Access Required for Individual DEC/Test Manager Commands

Command	Library Directory	Library Control File	Collection Subdirectories	Library Files	Collection Files
SHOW					
ALL	R	R			
BENCHMARK_DIRECTORY	R	R			
COLLECTION/FULL	R	R	R	R	R
HISTORY	R			R	
LIBRARY	R				
TEMPLATE_DIRECTORY	R	R			
Other SHOW commands	R	R			
STOP	RW	RW	RW		RW
SUBMIT	RW	RW	RW	RW	RWD
VERIFY	R	R	R		R
VERIFY/RECOVER	RW	RW	RD	R	RWD

2.3 Creating a Library—CREATE LIBRARY

The two steps for creating a DEC/Test Manager library are as follows:

1. Create a directory with the DCL `CREATE/DIRECTORY` command
2. Use the DTM `CREATE LIBRARY` command to set up this directory as a DEC/Test Manager library.

NOTE

The directory that is the target of the `CREATE LIBRARY` command must be empty. This directory must also be a first through seventh level directory.

For example, to create a directory called `[PROJECT.DTMLIB]`, type

```

$ CREATE/DIRECTORY [PROJECT.DTMLIB]
$

```

To designate this directory as a DEC/Test Manager library and to prepare it for use, type

```
⌘ DTM CREATE LIBRARY [PROJECT.DTMLIB]
%DTM-S-CREATED, DEC/Test Manager library DRA1: [PROJECT.DTMLIB] created
⌘
```

These two commands create the library [PROJECT.DTMLIB]. The CREATE LIBRARY command performs an implicit SET LIBRARY command so that you can now use DEC/Test Manager commands with this library.

When you create a DEC/Test Manager library, DEC/Test Manager sets up the logical name DTM\$LIB, which points to the directory you specify in the CREATE LIBRARY command. The definition remains in effect until you enter a SET LIBRARY command for a different directory, enter another CREATE LIBRARY command, or until you log off the system.

2.4 Converting a Library—**CONVERT LIBRARY**

If you have a DEC/Test Manager library that was created before Version 2.0, you must create a converted copy of it for use with this version of DEC/Test Manager. Use the DTM CONVERT LIBRARY command to do this. The CONVERT LIBRARY command converts everything in the library except collections.

NOTE

Your old library is not altered in any way and it exists as is until you delete it.

To convert a library, first create a new, empty directory to contain the converted library. Then enter the directory specification for the old (existing) DEC/Test Manager library followed by the directory specification for the new (converted) library.

For example, to create a directory for the converted library called [PROJECT.DTMV2LIB], type

```
⌘ CREATE/DIRECTORY [PROJECT.DTMV2LIB]
⌘
```

To convert [PROJECT.DTMLIB] and place the converted library in [PROJECT.DTMV2LIB], type

```
§ DTM CONVERT LIBRARY [PROJECT.DTMV1LIB] [PROJECT.DTMV2LIB]
%DTM-S-CONVERTED, version 1 library successfully converted to version 2
§
```

These two commands convert your old DEC/Test Manager library for use with this version of DEC/Test Manager. Enter the SET LIBRARY command to select the new library. For more information on the SET LIBRARY command, see Section 2.5.

2.5 Selecting an Existing Library

After you create a DEC/Test Manager library, you can use the SET LIBRARY command during a DEC/Test Manager session to select that library as the current library. You can use this command to begin a work session or to reference another DEC/Test Manager library. Subsequent DEC/Test Manager commands refer to this library until you select or create another library, or until you log off the system.

For example, to access the [PROJECT.DTMLIB] library, type

```
§ DTM SET LIBRARY [PROJECT.DTMLIB]
%DTM-S-LIBIS, DEC/Test Manager library is DRA1:[PROJECT.DTMLIB]
§
```

When you enter the SET LIBRARY command, DEC/Test Manager redefines the logical name DTM\$LIB to point to the directory you specify. This definition for DTM\$LIB remains in effect until you enter a SET LIBRARY command for a different directory, enter a CREATE LIBRARY command, or until you log off the system.

The SHOW LIBRARY command displays the directory specification of the current DEC/Test Manager library. If you have not defined a library with the CREATE LIBRARY or SET LIBRARY command, or if a library has become invalid, DEC/Test Manager issues an error message.

If you usually work with only one library, you may want to include a DTM SET LIBRARY/NOVERIFY command in your login command file. Because the SET LIBRARY/NOVERIFY command does not lock the DEC/Test Manager library, this command is preferable if someone may be using the library when you log in.

If you want to select a library quickly, without verifying that it is a valid DEC/Test Manager library, enter the SET LIBRARY/NOVERIFY command. You must still specify a valid DEC/Test Manager library.

Organizing a Test System

This chapter describes how to organize a test system, including information on the following:

- Creating tests and identifying them to DEC/Test Manager with test descriptions
- Associating template, benchmark, prologue, and epilogue files with test descriptions
- Displaying benchmark files and SESSION files for interactive tests
- Modifying, copying, displaying, and deleting test descriptions
- Storing files outside the DEC/Test Manager library
- Storing files in DEC/CMS libraries

To use DEC/Test Manager, you must first create a library, create the tests, and identify those tests to DEC/Test Manager.

3.1 Test Descriptions

A test description identifies a test and its related files to DEC/Test Manager. Each test requires a test description that contains the information DEC/Test Manager needs to run that particular test. The test description for a noninteractive test is created with the CREATE TEST_DESCRIPTION command, independent of both the creation of the test and its template file. The test description for an interactive test is created with the CREATE TEST_DESCRIPTION/RECORD or CREATE TEST_DESCRIPTION/INPUT command usually at the same time that the interactive terminal session is recorded in the test's template file.

A test description consists of **fields** for which you provide values. Table 3-1 summarizes the functions and legal values of the test description fields.

The test-name parameter is the only required information when you create a test description. All qualifiers are optional. If you do not provide a template name, DEC/Test Manager provides a default name of the form test-name.COM for a noninteractive test or test-name.SESSION for an interactive test, and places this name in the template field of the test description. If you do not supply a name for the benchmark file, DEC/Test Manager provides a default file name of the form test-name.BMK.

You can modify test descriptions during testing. For example, after running a test once, you may want to add an epilogue file as a filter. For more information about modifying test descriptions, see Section 3.6.

Table 3-1: Test Description Fields

Field	Field Value	Function
Test Name (Required)	test-name	Identifies the test description
Template (Optional)	file-spec	Points to the VAX/VMS command file that runs a noninteractive test, or points to the SESSION file containing an interactive test
Benchmark (Optional)	file-spec	Points to the file that contains the expected test output
Test Prologue (Optional)	file-spec	Points to a file that runs immediately before the template file
Test Epilogue (Optional)	file-spec	Points to a file that runs immediately after the template file
Variables (Optional)	variable-name and variable-value	Identifies the variables used with the template, prologue, or epilogue files for this test
Groups (Optional)	group-name	Identifies the groups to which the test description belongs
Filters (Optional)	filter-name	Identifies one or more filters to remove run-time variables from the result file
Remark (Optional)	quoted string	A comment you supply that is recorded in the history file

3.1.1 The Template File

For a noninteractive test, a template file is a DCL command file that you create to run a specified test. The only restrictions on the kinds of commands, programs, or actions you can use in a template file are that the template file must be able to execute as a batch job and it must generate some kind of output that DEC/Test Manager can compare. You may find it useful to name all template files you create for noninteractive tests with a standard file type, such as COM or TEST, so that they are easily identifiable. Template file names that DEC/Test Manager creates for noninteractive tests have the COM file type by default.

For an interactive test, DEC/Test Manager creates a template file when the interactive terminal session is recorded. This template file contains the interactive terminal session. Template file names that DEC/Test Manager creates for interactive tests have the SESSION file type by default.

A template file can be generalized to run more than one test. You may find this useful if you are running several tests that share common characteristics. This type of template file can be created with variables. A **variable** is a user-defined VAX/VMS symbol or logical name that DEC/Test Manager stores for use in running tests. Variables provide a convenient way to tailor a template file so that you can run more than one set of tests with the same file. Chapter 6 explains how to use variables in template files.

3.1.2 The Benchmark File

A benchmark file contains expected test results. It is the standard against which DEC/Test Manager compares the results of a test run. You can store benchmark files in the DEC/Test Manager library, in another directory, or in a DEC/CMS library. Specifying DTM\$LIB as the value for the /BENCHMARK qualifier on the CREATE TEST_DESCRIPTION and MODIFY TEST_DESCRIPTION commands causes benchmark files to be stored in the DEC/Test Manager library. Chapter 5 describes how to create benchmark files from test results.

3.1.3 The Test Prologue File

A test prologue file is a command file that you create. It is associated with one or more specified test descriptions and runs just before the test template file runs. The test prologue file can be used as a setup file to establish any special environment the test requires. No output from the test prologue file appears in the test results. Chapter 6 explains how to use test prologue files.

3.1.4 The Test Epilogue File

A test epilogue file is a command file that you create. It is associated with one or more specified test descriptions and runs just after the test template file runs. The test epilogue file can be used, for example, to perform clean up operations or to filter the result file of run-dependent data. Unlike the prologue file, the test epilogue file can directly alter the test results. Chapter 6 explains how to use test epilogue files.

3.2 Creating Noninteractive Tests

The three steps to create a noninteractive test and to identify it to DEC/Test Manager are as follows:

1. Write the test.
2. Write the template file.
3. Enter the `CREATE TEST_DESCRIPTION` command to create a test description for the test.

3.2.1 Writing the Test and the Template File

If you do not already have a test, invoke a text editor and write the test. Noninteractive tests are always created outside of DEC/Test Manager. A simple test of the DSR left margin command (LEFTMARGIN1.RNO) is shown in the following example:

```
This line should appear at the default left margin (0)
.lm15
This line should appear 15 spaces from the default left margin
.lm+10
This line should appear 25 spaces from the default left margin
.lm-20
This line should appear 5 spaces from the default left margin
```

For a noninteractive test, the template file is a VAX/VMS command file that runs the specified test. For more information about command procedures, see the *Guide to Using DCL and Command Procedures on VAX/VMS*. The following example shows the template file LMTEMPLATE.COM used to run the test LEFTMARGIN1.RNO:

```
$ ! TEMPLATE for a simple test of the DSR left margin (.LM) command
$ !
$ ! Display the input file
$ !
$ !
$ TYPE DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ !
$ ! Run DSR on the left margin test
$ !
$ !
$ DSR DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ !
$ ! Show the output of this runoff test
$ !
$ !
$ TYPE LEFTMARGIN1.MEM
$ !
$ !Clean up the temporary data file
$ !
$ DELETE DRA1:[PROJECT.RNO]LEFTMARGIN1.MEM;1
```

This template file runs DSR over the test file LEFTMARGIN1.RNO. To make the input file and the resulting MEM output file part of the test run record, both files are typed to SYS\$OUTPUT. This record of the test run (the LOG file) becomes the result file.

3.2.2 Creating the Test Description—CREATE TEST_DESCRIPTION

You create the test description for a noninteractive test with the CREATE TEST_DESCRIPTION command. The format is

```
DTM CREATE TEST_DESCRIPTION test-name [remark]
```

The test-name identifies a unique name for the test description. You supply a name that indicates the function of the test. For example, a test description for a test that adds numbers might be named TOTAL.

Information for test description fields is added with command qualifiers. Use the qualifiers described here when creating test descriptions for noninteractive tests.

The following qualifiers identify the file specifications for the template and benchmark files, and for the test prologue and epilogue files associated with the test:

```
/BENCHMARK  
/EPILOGUE  
/PROLOGUE  
/TEMPLATE
```

If you include the /BENCHMARK or /TEMPLATE qualifiers without a file specification or if you omit the qualifiers, DEC/Test Manager provides these defaults for the test description: test-name.BMK for the benchmark file and test-name.COM for the template file.

The following qualifiers associate filters and local variables with the test description:

```
/FILTER  
/[NO]LOG  
/VARIABLE
```

The /FILTER qualifier associates filters with the test description. These filters remove variable run-time information from the result file produced when the test is executed. The /VARIABLE qualifier associates local variables with the test description.

In the following example, the CREATE TEST_DESCRIPTION command creates the test description LMTEST1 and associates with it the FILE_NAMES filter:

```
⌘ DTM CREATE TEST_DESCRIPTION LMTEST1 /FILTER=(FILE_NAMES )-  
_⌘ "simple test of left margin command"
```

Because no template or benchmark file specifications are provided, DEC/Test Manager supplies its defaults: LMTEST1.COM as the file specification in the template field and LMTEST1.BMK as the file specification in the benchmark field.

Note that when DEC/Test Manager creates a test description, it checks the file specifications for correct syntax. It does not check whether the files exist.

3.3 Creating Interactive Tests

The two steps to create an interactive test with DEC/Test Manager are as follows:

1. Enter the CREATE TEST_DESCRIPTION/RECORD command to record an interactive terminal session and to create a test description for it.
2. Execute the interactive terminal session, which DEC/Test Manager records in the template file. Enter the designated termination character twice to terminate the interactive terminal session. The default termination character is CTRL/P.

You can also create an interactive test directly with an editor or by recording an interactive terminal session from an INPUT file. The information you need to do this is described in Appendix B. When you create a test description to identify this interactive test to DEC/Test Manager, use the /INTERACTIVE qualifier to mark the test description as describing an interactive test.

If DEC/Test Manager does not recognize the device type of terminal you are using to record the interactive terminal session, DEC/Test Manager will treat the terminal as a VT100-compatible terminal. The significance of device type and terminal characteristics when recording an interactive terminal session are described in Appendix B.

DEC/Test Manager supports screen sizes other than 24 lines by 80 columns. If your interactive terminal session will achieve a screen size larger than 24 lines by 132 columns, set the terminal screen to the largest size it will achieve before beginning the recording session.

3.3.1 Recording an Interactive Test and Creating its Test Description—CREATE TEST_DESCRIPTION/RECORD

The CREATE TEST_DESCRIPTION/RECORD command records an interactive terminal session (an interactive test), creates a test description for the test, and marks the test description as describing an interactive test. The interactive terminal session is recorded in the test template file. If you do not specify a file specification with the /TEMPLATE qualifier, DEC/Test Manager supplies the default, test-name.SESSION. The format is

```
DTM CREATE TEST_DESCRIPTION /RECORD test-name [remark]
```

The test-name identifies a unique name for the test description.

Information for test description fields is added with command qualifiers. When recording an interactive terminal session, you must include the /RECORD or /INPUT qualifier. The /INPUT qualifier is described in Appendix B.

The following qualifiers identify the file specifications for the template and benchmark files, and for the prologue and epilogue files associated with the test:

```
/BENCHMARK  
/EPILOGUE  
/PROLOGUE  
/TEMPLATE
```

If you include the /BENCHMARK or /TEMPLATE qualifiers without a file specification or if you omit the qualifiers, DEC/Test Manager provides these defaults for the test description: test-name.BMK for the benchmark file and test-name.SESSION for the template file.

The following qualifiers associate filters and local variables with the test description:

```
/FILTER  
/[NO]LOG  
/VARIABLE
```

The /FILTER qualifier associates filters with the test description. These filters remove variable run-time information from the result file produced when the test is executed. The /VARIABLE qualifier associates local variables with the test description.

The following qualifiers are used when you are recording an interactive terminal session:

```
/[NO]AUTO_COMPARE  
/COMPARISON_TYPE  
/INPUT  
/INTERACTIVE  
/RECORD  
/TERMINATION_CHARACTER
```

The `/AUTO_COMPARE` and `/NOAUTO_COMPARE` qualifiers determine whether automatic screen compare is enabled at the start of your recording session. By default, automatic screen compare is in effect. The `/NOAUTO_COMPARE` qualifier disables automatic screen compare. Automatic screen compare is described in Section 3.3.1.1. The `/COMPARISON_TYPE` qualifier determines how comparisons will be performed when the test is executed: screen by screen, record by record, or character by character.

The `/INPUT` qualifier specifies the file specification for an `INPUT` file from which a session file is to be recorded. DEC/Test Manager initiates an interactive terminal session and reads all input from this file until the file is exhausted. The procedure for generating a `SESSION` file from an `INPUT` file is described in Appendix B.

The `/INTERACTIVE` qualifier marks a test description as describing an interactive test. The `/RECORD` qualifier indicates that you want to record an interactive terminal session and create a test description for it. The `/TERMINATION_CHARACTER` qualifier specifies a new termination character. Specify a different termination character if the application you are testing interprets `CTRL/P`. Termination characters are discussed in Section 3.3.1.1.

3.3.1.1 Functions Available While Recording an Interactive Test

While you are recording an interactive terminal session, DEC/Test Manager provides you with a way to either temporarily suspend the recording session and enter a recording function or terminate the recording session. By using recording functions you can, for example, mark specific screens for inclusion in the benchmark file, or you can abort the recording session without saving the `SESSION` file and without creating or modifying the test description.

To perform a recording function, enter the termination character followed by another valid character. In this section, CTRL/P is assumed to be the termination character because CTRL/P is the default termination character. If you enter an illegal character after a CTRL/P—that is, a character other than one of the characters discussed here—the terminal bell sounds and no action is performed until after you reenter the command correctly.

Output generated by the recording functions—such as help text and prompts—will not appear in the SESSION or benchmark files. Table 3-2 lists the valid recording functions.

Table 3-2: Functions Available While Recording an Interactive Terminal Session

Command	Definition
CTRL/P B	(BEGIN_COMPARE) Starts automatic screen compare and terminates manual screen compare.
CTRL/P C	(COMPARE_SCREEN) Marks a screen for inclusion in the benchmark file (while in manual screen compare).
CTRL/P D	(DELAY) Allows you to specify a delta time which DEC/Test Manager is to wait at this point when the SESSION file you are recording is played or executed. This function is the same as the WAIT function.
CTRL/P E	(END_COMPARE) Terminates automatic screen compare and starts manual screen compare.
CTRL/P I	(INSERT) Inserts an INPUT file into the SESSION file you are recording. DEC/Test Manager prompts you for the input file specification.
CTRL/P W	(WAIT) Allows you to specify a delta time which DEC/Test Manager is to wait at this point when the SESSION file you are recording is played or executed. This function is the same as the DELAY function.
CTRL/P !	(BEGIN_COMMENT) Puts you in line editing mode so you can enter a comment into the SESSION file. Terminate the comment by pressing CTRL/Z.
CTRL/P ?	(HELP) Displays the current mode of screen compare and lists the available recording functions.

Table 3-2 (Cont.): Functions Available While Recording an Interactive Terminal Session

Command	Definition
CTRL/P CTRL/C	(ABORT) Aborts the recording session, does not save the SESSION or benchmark files, and does not create (or modify) the test description.
CTRL/P CTRL/P	(TERMINATE) Terminates the recording session, saves the SESSION or benchmark files, and creates (or modifies) the test description.
CTRL/P CTRL/Z	(QUIT) Terminates the recording session, saves the SESSION file, creates (or modifies) the test description, but does not save the benchmark file.

When you begin to record an interactive terminal session, automatic screen compare is in effect unless you specify otherwise with the /NOAUTO_COMPARE qualifier. Each time the terminal device is queued for input during the recording session, the current screen is included in the benchmark file and is marked for comparison with the test output when the test is run.

Pressing CTRL/P followed by E turns off automatic screen compare. While automatic screen compare is turned off, you must manually select screens for inclusion in the benchmark file and for comparison with the test output when the test is executed. To mark a screen for comparison, press CTRL/P followed by C at a point when the application you are testing is expecting input. The screen as it is displayed then is included in the benchmark file and marked for comparison when the test is executed. Pressing CTRL/P followed by B turns off manual screen compare and begins automatic screen compare.

Pressing CTRL/P twice terminates the recording session, saves the SESSION and benchmark files, and creates (or modifies) the test description. Pressing CTRL/P followed by CTRL/Z terminates the recording session, saves the SESSION file, and creates (or modifies) the test description, but does not save the benchmark file. Pressing CTRL/P followed by CTRL/C aborts the recording session without saving the SESSION or benchmark files and without creating (or modifying) the test description.

Pressing CTRL/P followed by a question mark (?) displays the current mode of screen compare (manual or automatic) and lists the available commands. Remove this help display by entering the command that refreshes the screen for the application you are testing. Many applications accept either CTRL/W or CTRL/R to refresh the screen.

Pressing CTRL/P followed by an exclamation point (!) puts you in line editing mode so that you can enter a comment in the SESSION file. Press CTRL/Z to terminate the comment. Press CTRL/C to abort the comment.

Pressing CTRL/P followed by D or W allows you to enter a delta time which DEC/Test Manager is to wait when the SESSION file is executed or played. If you enter an invalid time value, DEC/Test Manager rings the terminal bell. The D (or W) record is not entered in the SESSION file. Do not specify long DELAY and WAIT values. After the wait period has started, it cannot be aborted.

Pressing CTRL/P followed by I allows you to insert an INPUT file into the SESSION file. DEC/Test Manager prompts you for the file specification of the INPUT file and then takes all input from the INPUT file. When the INPUT file is exhausted, DEC/Test Manager returns control to the terminal. See Appendix B for more information about INPUT files.

3.3.1.2 Restrictions on Interactive Testing

There are certain restrictions on the types of interactive applications you can test using DEC/Test Manager. These restrictions are described in the following sections.

Timing-Dependent Applications

You cannot test timing-dependent applications. DEC/Test Manager cannot run the test at the same speed at which it was recorded; therefore, the screens marked for comparison will usually not match and the comparison of the test results will usually be unsuccessful. For example, you cannot test the VAX/VMS Phone Utility because your input as the person initiating the call depends on the person you are calling to answer your call. These two time-dependent events cannot be consistently duplicated.

Another example includes applications which either submit a batch job for processing or execute a subprocess while you have the choice of either waiting for the batch job or subprocess to finish or performing other operations with the application. The VAX Language-Sensitive Editor COMPILE command is an example of this type of timing-dependent application.

CTRL/Y

While recording a test, you cannot press CTRL/Y except at a point where the application being tested is expecting input. Pressing CTRL/Y at a point other than where the application is expecting input terminates the application at a random point which generally will not be duplicated when you run the test. For example, if while testing a DCL command such as the DIRECTORY command, you press CTRL/Y while command output is being displayed, you will create a screen that cannot be consistently duplicated and the comparison of the test results will usually be unsuccessful.

Type-Ahead

Use care when testing applications that depend on type-ahead. DEC/Test Manager does not support the testing of interactive applications that behave differently when type-ahead is present. Recorded terminal sessions are always played back without type-ahead. Thus, if you record a test in the presence of a type-ahead, DEC/Test Manager will play it back without type-ahead and the comparison of the test results will usually be unsuccessful.

If you must test an application built on a tool which uses type-ahead (such as the VAX Text Processing Utility or the VAX Language-Sensitive Editor), set your terminal to /NOTYPEAHEAD before you record the interactive terminal session. You do not need to set your terminal to /NOTYPEAHEAD when you play back the recorded terminal session. You do not need to set your terminal to /NOTYPEAHEAD when reading from an INPUT file.

Applications That Accept Unsolicited Input

You cannot use DEC/Test Manager to test applications that accept input without prompting for it. For example, the VAX/VMS Monitor Utility (MONITOR) cannot be tested with DEC/Test Manager. MONITOR displays screen after screen of continuously changing statistical information about the system. After you invoke MONITOR, it does not prompt you for input. It displays information until you terminate it by pressing CTRL/Z. The termination occurs in a way that cannot be consistently duplicated; thus, the comparison of the test results will usually be unsuccessful.

Applications That Use REGIS Graphics

You cannot use DEC/Test Manager to test screen images (graphics) generated by applications using REGIS instructions. The terminal simulator DEC/Test Manager uses to build, compare, and display screen images does not recognize REGIS instructions. Thus, DEC/Test Manager cannot compare screen images composed of REGIS instructions. Since DEC/Test Manager does capture the REGIS instructions, you can use DEC/Test Manager to verify that the REGIS instructions are being consistently transmitted by specifying that DEC/Test Manager perform character by character comparison when you create or modify the test.

VAXstations with UIS

You can use DEC/Test Manager to test applications on a VAXstation, provided that the applications use the normal VAX/VMS terminal I/O subsystem for all input and output—the application being tested operates within a DEC terminal emulation window. You cannot use DEC/Test Manager to test applications which make direct calls to the VAXstation graphics services (UIS).

3.3.1.3 Example Interactive Terminal Session

The following example shows the recording of an interactive terminal session. The file is an RNO file, which is a template for creating memos using DSR. It is recorded in the file MEMO_TEMPLATE.RNO.

```
.PS60,70
.FLAGS BOLD
.FLAGS SUBSTITUTE
.B
.NF
.TS28
      ^*I-N-T-E-R-O-F-F-I-C-E  M-E-M-O-R-A-N-D-U-M\*
.TS39,53.LM39
.B2
^*To:\*
^*Date:\*
^*From:\*
^*Ext:\*
.TS20.LM0.B3
^*Subject:\*
.B2
.F.J

.! Input the text of your memo starting on the next line.
```

To use the memo template to create a memo, do the following:

1. Edit a copy of the memo template to include the specific information for your memo.
2. Process this file with DSR to create a MEM file.
3. Print the MEM file as your completed memo.

To create a test to verify that the memo template file works properly, enter the CREATE TEST_DESCRIPTION/RECORD command and include the test-name (MEMO_TEST) and any remark you want to associate with the test description. At the DCL prompt, invoke a text editor (EDT is used in this example) to create the file MEMO.RNO. Copy the memo template file (MEMO_TEMPLATE.RNO) into the editing session and change to screen mode editing.

```

$ DTM CREATE TEST_DESCRIPTION/RECORD memo_test "Testing the DSR memo template"
%DTM-I-DEFAULTED, benchmark file name defaulted to MEMO_TEST.BMK
%DTM-I-DEFAULTED, template file name defaulted to MEMO_TEST.SESSIO
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.
```

```

$ EDIT/EDT memo.rno
Input file does not exist.
[EOB]
* INCLUDE memo_template.rno
* CHANGE
```

The file MEMO_TEMPLATE.RNO displays on the screen. Now fill in the required information for your memo. Everything you type is recorded in the MEMO_TEST.SESSIO file. When you are finished editing, leave the editor and save MEMO.RNO by pressing CTRL/Z and typing the EXIT command.

```

.PS60,70
.FLAGS BOLD
.FLAGS SUBSTITUTE
.B
.NF
.TS28
^*I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M^*
.TS39,53.LM39
.B2
^*To:\* John Dow
^*Date:\* July 20, 1986
^*From:\* Sally Smith
^*Ext:\* 1234
.TS20.LM0.B3
^*Subject:\* The new memo template.
.B2
.F.J
```

.! Input the text of your memo starting on the next line.
This is the new memo template for creating memos using
RUNOFF. Let me know how you like using it.

~Z
[EOB]

* EXIT
DRA1:[PROJECT]MEMO.RNO;1 29 lines

§

Enter the RUNOFF MEMO.RNO command to process the file and to
create a MEM file, then type the MEMO.MEM file. The output from
the RUNOFF command verifies whether the memo template is working
correctly.

§ RUNOFF MEMO.RNO
§ TYPE MEMO.MEM

I-N-T-E-R-O-F-F-I-C-E M-E-M-O-R-A-N-D-U-M

To: John Dow
Date: July 20, 1986
From: Sally Smith
Ext: 1234

Subject: The new memo template.

This is the new template for creating memos using epilogue RUNOFF. Let me
know how you like using it.

§

Finally, delete both the MEMO.RNO and MEMO.MEM files, terminate
the interactive terminal session by entering the termination character twice
(CTRL/P by default), and save the benchmark file.

§ DELETE MEMO.RNO;1
§ DELETE MEMO.MEM;1
§

~P

%DTM-I-BMK_SAVED, benchmark has been saved in file
DRA1:[PROJECT.DTMLIB]MEMO_TEST.BMK;1
%DTM-S-RECORDED, test MEMO_TEST has been successfully recorded in file
DRA1:[PROJECT]MEMO_TEST.SESSION
%DTM-S-CREATED, test description MEMO_TEST created
§

3.4 Displaying Benchmark Files for Interactive Tests—DISPLAY/BENCHMARK

You can display the benchmark file for an interactive test screen by screen by using the DISPLAY/BENCHMARK command. The format is

```
DTM DISPLAY/BENCHMARK test-name
```

The test-name parameter specifies the interactive test whose benchmark file you want to display. The /BENCHMARK qualifier, which is the default, specifies that you want to display the benchmark file for an interactive test.

You may want to display a benchmark file for an interactive test to verify that it reflects the expected results of an interactive terminal session. This would ensure that the benchmark file contains the correct results against which to test your interactive application. You may also want to display a benchmark file to verify the screens that you marked manually for inclusion in the benchmark file during an interactive terminal session.

When you enter the DISPLAY/BENCHMARK command, banner screen (Screen 0) displays as follows:

Bmk
Screen 0

```
DEC Test Manager
INTERACTIVE DISPLAY
Version 2.0

Type PF2 for help.
Use keypad 0 to move to next screen.
Type CTRL/Z to exit.

Test          MAIL_TEST
Benchmark file DTM$LIB:MAIL_TEST.BMK
```

ZK-4754-85

You can now do one of the following:

- Press PF2 to display HELP information.
- Press KP0 or RETURN to display the first screen of the benchmark file.
- Press CTRL/Z to terminate the DISPLAY/BENCHMARK command.

To move around in the benchmark file, you must press keypad keys. If your terminal does not have a keypad, the corresponding number keys give you the same movement; pressing CTRL/H displays help. The keypad keys and their associated functions are described in Table 3-3.

DEC/Test Manager displays the benchmark file screen by screen. The screen number display, initially on the top right corner of the screen, shows the number of the screen currently displayed.

You can move forward or backward one screen at a time by pressing KP0 or KP1, or you can return to the first screen by pressing KP5. Pressing PF2 displays HELP information. Pressing CTRL/W refreshes the screen. Pressing CTRL/Z terminates the display of the benchmark file.

Table 3-3: Key Definitions for DISPLAY/BENCHMARK

Keypad Key	Definition
KP0	Displays the next screen.
KP1	Displays the previous screen.
KP5	Displays the first screen in the benchmark file.
KP7	Toggles the screen number display. If the screen number display is on the screen, pressing KP7 removes it. If the screen number display is not on the screen, pressing KP7 displays it.
PF2	Toggles the HELP display. If the HELP display is on the screen, pressing PF2 removes it. If the HELP display is not on the screen, pressing PF2 displays it.
CTRL/W	Refreshes the screen.
CTRL/Z	Terminates display of the benchmark file.

The following command displays the benchmark file for the terminal session recorded in the previous example:

```
⌘ DTM DISPLAY MEMO_TEST /BENCHMARK
```

3.5 Running SESSION Files—PLAY

You can run the SESSION file for an interactive test with the PLAY command. A collection is not created and the results of the run are not saved. The PLAY command allows you to view your SESSION file while it runs in order to verify that it runs correctly.

If the terminal characteristics or terminal type for the display terminal differ from those for the recording terminal, the SESSION file may not appear as you expect it to. The format is

```
DTM PLAY file-spec
```

The file-spec parameter identifies the file specification for the SESSION file that is to be run.

The PLAY command takes the following qualifiers:

```
/[NO]DISPLAY_TERMINAL  
/[NO]LOG  
/[NO]RESULT_FILE
```

The `/[NO]DISPLAY_TERMINAL` qualifier determines whether the SESSION file is to display on a terminal. The `/[NO]RESULT_FILE` qualifier determines whether the output of the PLAY command is to be saved in a file.

The following command runs the SESSION file for the terminal session recorded in the previous example:

```
⚡ DTM PLAY MEMO_TEST.SESSION
```

By default, the SESSION file displays on `SY$OUTPUT` and the output is not saved in a file.

3.6 Modifying Test Descriptions—MODIFY TEST_DESCRIPTION

You can modify the information in test description fields by using the `MODIFY TEST_DESCRIPTION` command. Test descriptions for both interactive and noninteractive tests can be modified. The format is

```
DTM MODIFY TEST_DESCRIPTION test-group-expression [remark]
```

The `test-group-expression` selects the test descriptions to modify. Valid items in a `test-group-expression` are valid test-expressions and group-expressions. You can select tests by test-name, by a wildcard form of such a name, by group-name, or by a wildcard form of such a name. The qualifiers for the `MODIFY TEST_DESCRIPTION` command parallel the qualifiers for the `CREATE TEST_DESCRIPTION` command, with the addition of negated qualifiers. The positive qualifiers change the indicated test description field values. The negated qualifiers remove the indicated field values from the test description.

The following qualifiers replace the file specification for the template file, benchmark file, test prologue file, or test epilogue file with the indicated file specification:

```
/[NO]BENCHMARK  
/[NO]EPILOGUE  
/[NO]PROLOGUE  
/[NO]TEMPLATE
```

The `/NOBENCHMARK` and `/NOTEMPLATE` qualifiers cause those test description fields to contain their default values. The `/NOPROLOGUE` and `/NOEPILOGUE` qualifiers remove the prologue and epilogue file specification from the test description.

The following qualifiers replace or remove filters, local variables, and the remark from the test description you are modifying:

```
/[NO]FILTER  
/[NO]INTERACTIVE  
/[NO]LOG  
/[NO]REMARK  
/[NO]VARIABLE
```

The `/[NO]FILTER` qualifier associates or removes the specified filters from the test description. These filters remove variable run-time information from the result file that is produced when the test is executed. The `/INTERACTIVE` qualifier marks a test description as describing an interactive test. The `/NOINTERACTIVE` qualifier marks a test description as describing a noninteractive test. The `/[NO]REMARK` qualifier associates or removes the remark from the test description. The `/[NO]VARIABLE` qualifier associates or removes local variables from the test description.

The following qualifiers are used when you modify the test description for an interactive test:

```
/[NO]AUTO_COMPARE  
/COMPARISON_TYPE  
/INPUT  
/RECORD  
/TERMINATION_CHARACTER
```

The `/AUTO_COMPARE` and `/NOAUTO_COMPARE` qualifiers determine whether automatic screen compare is enabled at the start of your recording session. By default, automatic screen compare is in effect. The `/NOAUTO_COMPARE` qualifier disables automatic screen compare. Automatic screen compare is described in Section 3.3.1.1. The `/COMPARISON_TYPE` qualifier determines how comparisons are performed when the test is executed: screen by screen, record by record, or character by character. The `/RECORD` qualifier indicates that you want to record another interactive terminal session and to associate it with the test description you are modifying. The `/TERMINATION_CHARACTER` qualifier specifies a new termination character. Specify a different termination character if the application you are testing interprets CTRL/P. Termination characters are discussed in Section 3.3.1.1.

The `/INPUT` qualifier specifies the file specification for an INPUT file that is to be converted to a session file. DEC/Test Manager initiates an interactive terminal session and reads all input from this file until the file is exhausted. The procedure for generating a SESSION file from an INPUT file is described in Appendix B.

You may wish to modify a test description if, for example, after running a test you decide to add an epilogue file that filters test results. The following command adds an epilogue file to the test description for the test LMTEST1:

```
$ DTM MODIFY TEST_DESCRIPTION LMTEST1 /EPILOGUE=LMEPI.COM  
_Remark: "Adding a new test epilogue"
```

The following example removes the existing benchmark file specification for LMTEST1 and leaves the field empty. Note that if the benchmark file exists in the DEC/Test Manager library, it is deleted. If it exists outside the DEC/Test Manager library, it is not deleted. DEC/Test Manager issues informational messages that inform you of these conditions.

```
‡ DTM MODIFY TEST_DESCRIPTION LMTEST1 /NOBENCHMARK  
_Remark: "Removing the benchmark file"
```

If you include the /BENCHMARK qualifier but do not specify a file specification, DEC/Test Manager supplies the default file name test-name.BMK. The following example includes the /RECORD qualifier to modify a test description by recording a new interactive terminal session over the existing template file. This command also marks the modified test description as an interactive test.

```
‡ DTM MODIFY TEST DESCRIPTION STOPEST1 /RECORD "Recording the STOP test again"
```

3.7 Copying Test Descriptions—COPY TEST_DESCRIPTION

Use the COPY TEST_DESCRIPTION command to create an exact or modified copy of a test description in the DEC/Test Manager library. The COPY TEST_DESCRIPTION command allows you to create a series of similar test descriptions without repeatedly needing to enter the CREATE TEST_DESCRIPTION command and to fill in all the field values. The format is

```
DTM COPY TEST_DESCRIPTION test-name1 test-name2 [remark]
```

The test-name1 parameter identifies a test description existing in the current DEC/Test Manager library. Because test names must be unique, test-name2 (the name of the new test description) must be a name that you are not already using for a test description in the current DEC/Test Manager library. You cannot use wildcards with this command.

The positive qualifiers for the COPY TEST_DESCRIPTION command allow you to change the value for a test description field. The negated qualifiers, except /NOTEMPLATE, allow you to remove a value from a test description field. The /NOTEMPLATE qualifier replaces the current template file name with one of the form test-name.COM for a noninteractive test and test-name.SESSION for an interactive test. You must also either copy the test description with its filter, variable, and group field values intact or eliminate the field values altogether; you cannot modify these values when you copy the test description. Qualifiers that take values replace the current value for the corresponding field with the new

value you specify. If you do not specify a qualifier, the value for that field is copied from the existing test description to the new test description. If the test description you are copying identifies an interactive test, the new test description is marked as interactive.

The following qualifiers replace the file specification for the template file, test prologue file, or test epilogue file with the indicated file specification or remove the indicated file specification from the test description:

```
/[NO]EPILOGUE  
/[NO]PROLOGUE  
/[NO]TEMPLATE
```

Note that you cannot change the value of the benchmark field. You must copy the benchmark associated with the existing test. In the case of the template file, the /NOTEMPLATE qualifier instructs DEC/Test Manager to use the default template file name.

The following qualifiers remove filters, groups, or local variables from the test description you are copying:

```
/NOFILTERS  
/NOGROUPS  
/NOVARIABLES
```

The new test description belongs to the same groups as the existing test description and is associated with the same filters and variables as the existing test description unless you specify the /NOFILTERS, /NOGROUPS, or /NOVARIABLES qualifiers.

The /REMARK and /NOREMARK qualifiers associate a new remark with the copied test description or remove the current remark from the copied test description. The COPY TEST_DESCRIPTION command also takes the /[NO]LOG qualifier.

For a test description describing an interactive test, the /COMPARISON_TYPE qualifier changes how comparisons are performed when the test is executed: screen by screen, record by record, or character by character.

For example, the following command copies the test description LMTEST1 to the test description LMTEST2. The command also removes the prologue file and associates the remark "Copied from LMTEST1 with no prologue" with the new test description.

```
‡ DTM COPY TEST_DESCRIPTION LMTEST1 LMTEST2 /NOPROLOGUE  
_Remark: "Copied from LMTEST1 with no prologue"
```

3.8 Displaying Test Descriptions—SHOW TEST_DESCRIPTION

You can display test descriptions by using the SHOW TEST_DESCRIPTION command. The format is

```
DTM SHOW TEST_DESCRIPTION test-group-expression
```

The test-group-expression selects the test descriptions to display. Valid items in a test-group-expression are valid test-expressions and group-expressions. You can select tests by test-name, by a wildcard form of such a name, by group-name, or by a wildcard form of such a name.

The SHOW TEST_DESCRIPTION command takes the following qualifiers:

```
/BENCHMARK  
/BRIEF  
/COMPARISON_TYPE  
/EPILOGUE  
/FILTER  
/FULL  
/GROUPS  
/INTERMEDIATE  
/NAME  
/OUTPUT  
/PROLOGUE  
/REMARK  
/TEMPLATE  
/TYPE  
/VARIABLE
```

The /BRIEF, /INTERMEDIATE, and /FULL qualifiers control the amount of information displayed. The default, /INTERMEDIATE, displays the contents of all test description fields except the groups and variables fields. The /BRIEF qualifier displays the test-name only. The /FULL qualifier displays the contents of all test description fields.

The /BENCHMARK, /EPILOGUE, /PROLOGUE, and /TEMPLATE qualifiers display the file specification for the specified output files.

The /FILTER, /GROUPS, and /VARIABLE qualifiers display the names of all filters, groups, and variables associated with the test description.

The /COMPARISON_TYPE, /NAME, and /REMARK qualifiers display the comparison type, test-name, and remark associated with the test description.

The /TYPE qualifier displays the type of test, interactive or noninteractive, associated with the test description.

The /OUTPUT qualifier directs the SHOW TEST_DESCRIPTION command output to the specified file.

The following command displays a test description for the test LMTEST1. The default /INTERMEDIATE qualifier is in effect.

```
⌘ DTM SHOW TEST_DESCRIPTION LMTEST1

Test descriptions in DEC/Test Manager Library DISK$USER01:[PROJECT.DTMLIB]

LMTEST1      "simple test of left margin command"
  Template : [PROJECT.COM]LMTEMPLATE.COM
  Benchmark: LMTEST1.BMK
  Prologue  : None Specified
  Epilogue  : None Specified
⌘
```

3.9 Deleting Test Descriptions—DELETE TEST_DESCRIPTION

Use the DELETE TEST_DESCRIPTION command to delete a test description from the DEC/Test Manager library. The format is

```
DTM DELETE TEST_DESCRIPTION test-expression [remark]
```

The test-expression parameter identifies the test descriptions to delete. The DELETE TEST_DESCRIPTION command takes the /[NO]CONFIRM and /[NO]LOG qualifiers.

If the test description has a benchmark file that resides in the DEC/Test Manager library, the benchmark file is also deleted. If the benchmark file is outside the library, it is unaffected. The DELETE TEST_DESCRIPTION command does not affect result files or differences files that may have been produced for this test during a collection run. Because these files are associated with the collection, they are deleted with the collection rather than with the test description.

The following command deletes the test description LMTEST1. Because the benchmark is in the DEC/Test Manager library, it also is deleted.

```
⌘ DTM DELETE TEST_DESCRIPTION LMTEST2 "Deleting the revised left margin test"
```

A test description cannot be deleted while it is a member of any group. Use the REMOVE TEST_DESCRIPTION command to remove a test description from all groups to which it belongs. Then use the DELETE TEST_DESCRIPTION command to delete the test description.

3.10 Storing Files Outside the DEC/Test Manager Library

Benchmark files can be stored either within the DEC/Test Manager library or outside of it. Template files for noninteractive tests, test epilogues and prologues, and collection epilogues and prologues cannot be stored in the DEC/Test Manager library. These files must be stored outside the DEC/Test Manager library. All files stored outside the DEC/Test Manager library (benchmarks, templates for interactive and noninteractive tests, and prologues and epilogues for both collections and tests) can be stored in either VAX/VMS directories or DEC/CMS libraries.

3.10.1 Establishing Default Directories for Benchmark and Template Files—SET BENCHMARK_DIRECTORY and SET TEMPLATE_DIRECTORY

You can specify a hierarchy of directories for benchmark and template files. These files can all be stored in the same directory or they can be stored in separate directories.

The SET BENCHMARK_DIRECTORY and SET TEMPLATE_DIRECTORY commands establish default benchmark and template directories for the current DEC/Test Manager library. If you do not specify default benchmark and template directories, DEC/Test Manager uses your current default directory (SYS\$DISK:[]) for template files and the DEC/Test Manager library (DTM\$LIB) for benchmark files.

NOTE

Using default benchmark and template directories rather than including a directory on each file specification makes processing of the commands that access these files faster.

The /BENCHMARK_DIRECTORY and /TEMPLATE_DIRECTORY qualifiers on the CREATE COLLECTION command override the default directories for the current DEC/Test Manager library (if defaults exist). For the collection being created, the specified directories are used for the benchmark and template files.

When you create or modify a test description, you can include a directory specification with the benchmark or template file specifications. DEC/Test Manager searches the specified directory for the files for that test rather than either the collection or library default directories. Thus, you can

ensure that DEC/Test Manager uses a special template or benchmark file for a new test or for a temporary test.

Remove a default template or benchmark directory without replacing it by entering the SET NOBENCHMARK_DIRECTORY or SET NOTEMPLATE_DIRECTORY command. These commands return the template and benchmark directories to their DEC/Test Manager defined default values. The template directory will then be SYS\$DISK[:] and the benchmark directory will be DTM\$LIB.

3.10.2 Using DEC/CMS Libraries for Benchmark and Template Files

DEC/Test Manager allows you to store your template and benchmark files in one or more DEC/CMS libraries. To do this, you must create the DEC/CMS libraries and identify them to DEC/Test Manager.

- Create a directory and make it a DEC/CMS library. This directory cannot be a subdirectory of the DEC/Test Manager library.

For information about setting up a DEC/CMS library, see the *VAX DEC/CMS Reference Manual*.

- Identify the DEC/CMS library to DEC/Test Manager by including the directory specification for the DEC/CMS library in the appropriate command. You can specify a DEC/CMS library on any command where you can specify a directory specification for a template or benchmark file.

DEC/Test Manager checks all directory specifications when template and benchmark files are specified to determine if they refer to a directory that is a DEC/CMS library. If a directory specification is included and it specifies a DEC/CMS library, DEC/Test Manager issues the appropriate DEC/CMS commands to access the files. If a file name is included, DEC/Test Manager interprets the file name as a DEC/CMS element name and accesses the appropriate DEC/CMS element. If a DEC/CMS class is also specified with the /CLASS qualifier on a CREATE COLLECTION command, DEC/Test Manager accesses the indicated generation of the element.

DEC/Test Manager issues the following DEC/CMS commands to access files in DEC/CMS libraries:

- CMS DIFFERENCES
- CMS FETCH
- CMS RESERVE

- CMS CREATE ELEMENT
- CMS REPLACE
- CMS INSERT/SUPERSEDE
- CMS SET LIBRARY
- CMS UNRESERVE

When you execute a collection, DEC/Test Manager performs the CMS FETCH command to retrieve the template element from the DEC/CMS library. DEC/Test Manager deletes the template file after using it.

When you compare a collection, DEC/Test Manager compares the result file with the specified benchmark element in the DEC/CMS library. If a DEC/CMS class is specified for the benchmark element, DEC/Test Manager compares the result file with the appropriate generation of the benchmark element.

When you update an existing benchmark file from the Review subsystem, DEC/Test Manager performs the CMS RESERVE command to retrieve and reserve the specified benchmark element from the DEC/CMS library. DEC/Test Manager then issues the CMS REPLACE command to replace the reserved benchmark element with the result file from the current test run. The new benchmark element, created from the result file, is no longer reserved.

When you create a new benchmark file for a test, DEC/Test Manager performs the CMS CREATE ELEMENT command to create a new benchmark element in the DEC/CMS library.

If you specify a DEC/CMS class for the benchmark file, DEC/Test Manager performs the CMS INSERT/SUPERSEDE command to place the result file in the appropriate benchmark generation. If you update a benchmark generation other than the latest, a variant line 'D' is created. If the variant line 'D' already exists, the update fails. If the reserve fails, DEC/Test Manager unreserves the benchmark element.

When you print or display a benchmark file from the Review subsystem with the PRINT/BENCHMARK or SHOW/BENCHMARK command, DEC/Test Manager performs the CMS FETCH command to retrieve the benchmark element from a DEC/CMS library. DEC/Test Manager deletes the benchmark file after printing or displaying it.

When you record an interactive test and the template and benchmark files are stored in DEC/CMS libraries, DEC/Test Manager performs the CMS CREATE ELEMENT command to create new elements for new benchmark and template files if the files do not already exist. If they do exist, DEC/Test Manager performs the CMS RESERVE and CMS REPLACE commands to access existing files.

3.10.3 Establishing Collection Prologue and Epilogue Files—SET PROLOGUE and SET EPILOGUE

The SET PROLOGUE and SET EPILOGUE commands establish default collection prologues and epilogues for all collections created in the current DEC/Test Manager library.

The /PROLOGUE and /EPILOGUE qualifiers on the CREATE COLLECTION command override the default for a particular collection. These qualifiers can specify another file in the same directory or another file in a different directory. The /NOPROLOGUE and /NOEPILOGUE qualifiers specify that no prologue or epilogue is to be run with a collection.

3.10.4 Establishing Test Prologue and Epilogue Files—CREATE TEST_DESCRIPTION/PROLOGUE and /EPILOGUE

The /PROLOGUE and /EPILOGUE qualifiers on the CREATE TEST_DESCRIPTION command associate a prologue or epilogue file with a particular test.

The /PROLOGUE and /EPILOGUE qualifiers on the MODIFY TEST_DESCRIPTION command replace the prologue or epilogue file for a specified test. These qualifiers can specify another file in the same directory or another file in a different directory. The /NOPROLOGUE and /NOEPILOGUE qualifiers specify that no prologue or epilogue file is to be run with this test.

3.10.5 Storing Prologue and Epilogue Files in DEC/CMS Libraries

If the directory specification portion of the file specification for either a collection or test prologue or epilogue file identifies a DEC/CMS library, DEC/Test Manager performs the DEC/CMS FETCH command on the latest generation of the specified prologue or epilogue file from the DEC/CMS library. You cannot specify DEC/CMS classes for prologue and epilogue files.

Selecting and Running Test Collections

This chapter describes how to select and run tests using DEC/Test Manager, including information on the following:

- Selecting tests to execute
- Creating collections
- Executing tests in batch
- Executing tests interactively
- Comparing test results
- Displaying the collection summary
- Deleting collections
- Recreating collections
- Stopping collections

The three steps to executing tests with DEC/Test Manager are as follows:

1. **Test selection.** In this step, you select a test description or any combination of test descriptions in the DEC/Test Manager library, place them in a collection, and give the collection a name that is unique in the current DEC/Test Manager library. DEC/Test Manager reads each test description in the collection and uses the files specified there to create the collection command file to run each test in the collection. You can include a test description in more than one collection.
2. **Test execution.** In this step, you either submit the collection for batch mode processing or run the collection interactively. Either way, DEC/Test Manager sets up the test environment and executes the collection command file it has created. The collection command file invokes the files for each test description in the collection. Each test

generates a separate result file. The result file contains the output generated by the template file. The result file is like the batch log file for the associated template command file minus the run summary information generally included in a batch log file. DEC/Test Manager stores the result files for all tests in a collection under the collection name. DEC/Test Manager then associates the result file with the test by adding its name to the **result description** for the test. Result descriptions are described further in Chapter 6.

NOTE

For the same test executed both interactively and in batch mode, the results may vary if your login command file sets up your interactive environment differently than it sets up your batch environment. For more information, see the *Guide to Using DCL and Command Procedures on VAX/VMS*.

3. **Results comparison.** In this step, DEC/Test Manager looks for a benchmark file for each test and compares the result file with the benchmark file, if the benchmark file exists. Any differences between the files are stored in a **difference file** produced by DEC/Test Manager. If the result file is identical to the benchmark file, DEC/Test Manager deletes the result file and the comparison status for the test is successful. If the result file and the benchmark file do not match, the comparison status is unsuccessful. If no benchmark file for the test exists, the test has the new test comparison status and no comparison is performed. Comparison status is described further in Chapter 5.

After looking at a result file, you may want to make it the benchmark file for the test. You can create or update benchmark files from result files with the Review subsystem UPDATE command. The Review subsystem is described further in Chapter 5. You can also create a benchmark file for an interactive test immediately after recording the interactive terminal session.

4.1 Creating Collections—CREATE COLLECTION

To create a collection, you do the following:

- Select one or more tests or groups of tests to execute
- Identify the selected tests with a unique collection-name

The CREATE COLLECTION command generates a set of tests and their related files. A collection consists of a “snapshot” of the specified test descriptions and the library at the time the collection is created. Thus, if you change a test description or the contents of a group after including it in a collection, those changes are not reflected in the collection unless you recreate it. Section 4.6 describes how to recreate existing collections. The format is

```
DTM CREATE COLLECTION collection-name test-group-expression [remark]
```

The test-group-expression selects the tests to be placed in the collection. Valid items in a test-group-expression are valid test-expressions and group-expressions. You can select tests by test-name, by a wildcard form of such a name, or by specifying a group name to indicate that all the test descriptions in the group should be part of the collection. Group-names and group-expressions in a test-group-expression must be labeled as naming a group with the parameter qualifier /GROUP.

The collection-name follows the same rules as the file name component of a VAX/VMS file specification. The collection-name cannot begin with the letters DTM\$.

The CREATE COLLECTION command takes the following qualifiers:

```
/[NO]BENCHMARK_DIRECTORY  
/[NO]VERIFY  
/CLASS  
/[NO]COMPARE  
/[NO]EPILOGUE  
/[NO]LOG  
/[NO]PROLOGUE  
/[NO]SUBMIT  
/[NO]TEMPLATE_DIRECTORY  
/VARIABLE
```

If you do not use the /SUBMIT qualifier, execute the collection with the DTM SUBMIT command or with the DTM RUN command. If you use the /NOCOMPARE qualifier, compare the collection results with the COMPARE command after the collection has executed. These commands and qualifiers are described further in Section 4.2.

By default, DEC/Test Manager will attempt to fully resolve file specifications for all required files including device, directory, file-name, and file-type before creating the collection. DEC/Test Manager must be able to locate every file, except the benchmark file, named in a test description. If one or more needed files are missing, DEC/Test Manager lists all the missing files and does not create the collection. If you specify a benchmark file that DEC/Test Manager cannot find, DEC/Test Manager creates the collection without the benchmark file and treats the test whose benchmark file is missing as a new test.

If you specify the /NOVERIFY qualifier, DEC/Test Manager creates the collection while resolving only those file specifications included in the CREATE COLLECTION command line.

Do not modify or delete any of the files pointed to by a test description while that test is included in a collection. Benchmark files are the only exception to this; they may be updated using the appropriate DEC/Test Manager commands. If you execute a collection containing an incomplete test, the collection run will proceed but the incomplete test will not run.

The following example creates a collection containing one test description and immediately submits it to the batch queue:

```
⌘ DTM CREATE COLLECTION TRIAL RMTEST1 /NOVERIFY /SUBMIT=(NOTIFY) /PROLOGUE=SETUP.COM-  
_⌘ Remark: First run of test RMTEST1
```

The first parameter identifies the collection name TRIAL. The second parameter, RMTEST1, identifies the test you want included in the collection and run. The /NOVERIFY qualifier specifies that DEC/Test Manager is to create the collection without verifying the existence of all files associated with the test description. The /SUBMIT qualifier specifies that DEC/Test Manager is to submit the collection to the batch queue as soon as the collection is created, and the /NOTIFY qualifier specifies that you will be notified when the batch job has completed. The /PROLOGUE=SETUP.COM qualifier specifies that SETUP.COM is the collection prologue. Because the collection prologue file SETUP.COM is specified on the CREATE COLLECTION command (it is associated with the collection), DEC/Test Manager verifies the existence of this file even if you specify the /NOVERIFY qualifier.

The following example creates a collection that executes all the tests in the group MARGINS:

```
Ⓕ DTM CREATE COLLECTION MARGINS_NOV MARGINS/GROUP /NOVERIFY-  
_Ⓕ "margin tests as of November"
```

The collection is named MARGINS_NOV and contains all the tests in the group MARGINS. The /GROUP positional qualifier indicates that MARGINS identifies a group rather than a test description. The /NOVERIFY qualifier indicates that DEC/Test Manager is to create the collection without verifying that all files associated with the test description exist. When DEC/Test Manager creates the collection, it includes the current version of all test descriptions currently in the group MARGINS. (You can display the complete contents of a group with the DTM SHOW GROUP/CONTENTS=ALL command.)

Because the /SUBMIT qualifier is not specified in this example, DEC/Test Manager does not execute the collection when it is created. Enter either the DTM SUBMIT command or the DTM RUN command to execute the collection. If when executing the collection DEC/Test Manager cannot find the test prologue file (if one is specified) or the template file associated with a test description, that test will not be executed. When the collection is compared, that test will have the not run comparison status. The DTM SUBMIT command is described in Section 4.2.1. The DTM RUN command is described in Section 4.2.2.

4.2 Executing Collections

When you execute a collection, DEC/Test Manager executes the collection command file. This command file does the following:

- Defines DTM\$COLLECTION_NAME
- Defines global variables
- Runs the collection prologue file
- For each test, calls a generic command file in the DEC/Test Manager library which performs the following:
 - defines local variables
 - defines DTM\$TEST_NAME
 - runs the test prologue
 - runs the test template
 - defines DTM\$RESULT

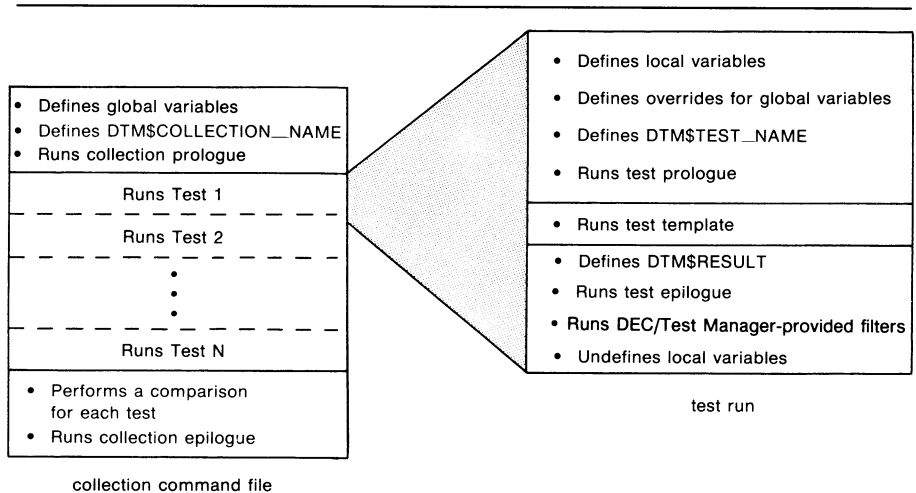
- runs the test epilogue
- runs DEC/Test Manager-provided filters
- undefines local variables
- undefines DTM\$RESULT
- Compares the result file with the benchmark file
- Runs the collection epilogue

DTM\$COLLECTION_NAME, DTM\$TEST_NAME, and DTM\$RESULT are variables defined by DEC/Test Manager that point to the collection-name, test-name, and result file, respectively, for the current collection. These variables are described in Chapter 6.

DEC/Test Manager executes collections two ways: in batch or interactively. The DTM SUBMIT and CREATE COLLECTION/SUBMIT commands both execute the collection by submitting it to the batch queue. The DTM RUN command executes the collection interactively. The same results will be generated in either mode, whether the tests in the collection are interactive or not.

When a collection executes, all the tests in the collection execute. After the collection has executed and has been compared, the test results can be examined with the Review subsystem. Figure 4-1 shows test execution using DEC/Test Manager.

Figure 4-1: Steps in Test Execution Using DEC/Test Manager



ZK-2081-84

4.2.1 Executing Collections in Batch—SUBMIT

There are two ways to execute a collection in batch:

1. Instruct DEC/Test Manager to submit the collection to the batch queue after creating it.
2. Submit the collection manually using the DTM SUBMIT command.

If you use the /SUBMIT qualifier with the CREATE COLLECTION command, DEC/Test Manager creates the collection and automatically submits it, using the default qualifiers to the DTM SUBMIT command. You can also optionally specify any of the DTM SUBMIT command qualifiers.

If you use the default qualifier `/NOSUBMIT` on the `CREATE COLLECTION` command, use the `DTM SUBMIT` command to execute the collection in batch. The format is

```
DTM SUBMIT collection_name
```

The collection-name specifies the collection to be submitted to the batch queue.

The `SUBMIT` command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]KEEP  
/[NO]LOG
```

The `DTM SUBMIT` command also takes the following `DCL SUBMIT` command qualifiers:

```
/AFTER           /CHARACTERISTICS   /CPUTIME  
/[NO]HOLD       /[NO]KEEP          /[NO]LOG_FILE  
/NAME           /[NO]NOTIFY        /[NO]PRINTER  
/PRIORITY       /QUEUE             /USER  
/WSDEFAULT      /WSEXTENT          /WSQUOTA
```

The default qualifiers to the `DTM SUBMIT` command correspond to those of the `DCL SUBMIT` command. Defaults are listed in the *VAX/VMS DCL Dictionary*. Note that you cannot use the `DCL SUBMIT` command to submit a collection. You must use the `DTM SUBMIT` command.

You can submit a collection more than once. If you attempt to resubmit a collection that you have executed and not reviewed, `DEC/Test Manager` prompts you to confirm that you want to resubmit the collection without reviewing it, unless you specify the `/NOCONFIRM` qualifier.

The following example submits the collection `MARGINS_NOV` to the batch queue. The `/NOTIFY` qualifier notifies you when the job is completed or aborted.

```
⚡ DTM SUBMIT /NOTIFY MARGINS_NOV
```

4.2.2 Executing Collections Interactively—RUN

Use the DTM RUN command to execute a collection interactively. The collection can contain interactive or noninteractive tests, or both.

Using the DTM RUN command to execute the tests in a large collection could tie up your terminal for a long time. Thus, it is better to use the DTM SUBMIT command to execute large collections. The results for both interactive and noninteractive tests are the same whether they are executed interactively or noninteractively, provided your batch and interactive environments are the same.

The format is

```
DTM RUN collection-name
```

The collection-name specifies the collection to be run. If you enter a test-name rather than an existing collection-name as the parameter to the DTM RUN command, DEC/Test Manager prompts you to create a collection. Collections created this way contain only the specified test and have the same collection-name as the test-name.

If you attempt to run a collection that you have executed before but not reviewed, DEC/Test Manager prompts you to confirm that you want to resubmit the collection without reviewing it, unless you specify the /NOCONFIRM qualifier.

The DTM RUN command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG  
/[NO]LOG_FILE  
/[NO]OUTPUT
```

By default, the DTM RUN command displays on your screen the contents of the result file for each test. If you want to see messages from your prologue or epilogue file, or from the automatic comparison, specify the /LOG_FILE qualifier.

The following example runs the collection LMTEST on your screen:

```
DTM RUN lntest "run of the left margin test"
```

You will see this message:

```
Starting LMTEST1 test run...
```

The output of the test run will follow. For example, if the collection LMTEST contains the noninteractive test LMTEST1, the output of the test run is as follows:

```
This line should appear at the default left margin (0)
.lm15
This line should appear 15 spaces from the default left margin
.lm+10
This line should appear 25 spaces from the default left margin
.lm-20
This line should appear 5 spaces from the default left margin
```

```
This line should appear at the default left margin (0)
                This line should appear 15 spaces from the default left
                margin
                        This line should appear 25 spaces from the
                        default left margin
                This line should appear 5 spaces from the default left margin

Performing post-run cleanup with comparison...
$
```

4.3 Comparing Test Results—COMPARE

For every test in a collection that runs to completion, by default DEC/Test Manager automatically compares the result file against the benchmark file (if it exists), records the comparison status for the test, and records any differences in a difference file. If the benchmark file and result file match, DEC/Test Manager deletes the result file.

Some of the tests in a collection may not run, either because the system crashes or because you stop execution of the collection with the DTM STOP command (or by pressing CTRL/C if you are using the DTM RUN command). When a collection is partially run, no automatic comparison is performed. Use the COMPARE command to manually compare the existing result files with the appropriate benchmark files. To execute the tests that did not run, you can recreate the collection and execute the entire collection again. To execute only the tests that did not run, from the Review subsystem you can use the INSERT command to include in a group only the tests that did not run, create a collection containing that group of tests, and execute the collection.

DEC/Test Manager reports the comparison status for each test as one of the following:

- Comparison aborted—for tests whose comparison could not be completed.
- New test—for tests that do not have a benchmark file.
- Not run—for tests that did not run in a partially run collection.
- Successful—for tests whose benchmark and result files match.
- Unsuccessful—for tests whose benchmark and result files do not match.
- Updated—for tests whose benchmark files have been updated after the comparison has been performed for a collection.

A comparison can be performed only once for each collection, and a collection must be compared before it can be reviewed. When you create a collection, specify whether you want the comparison to occur automatically, that is, immediately after the collection is executed. If you use the /NOCOMPARE qualifier with the CREATE COLLECTION command, the comparison step is not included when the collection is executed. Use the DTM COMPARE command to compare the current test results for each test with its benchmark file, if it exists.

The format is

```
DTM COMPARE collection-name
```

The collection-name specifies the collection to compare.

The DTM COMPARE command takes the following qualifiers:

```
/CHARACTERS  
/FULL  
/IGNORE  
/[NO]LOG  
/[NO]PARALLEL  
/RECORDS  
/SCREENS  
/WIDTH
```

The /CHARACTERS, /RECORDS, and /SCREENS qualifiers describe the type of comparison DEC/Test Manager is to perform. The /IGNORE qualifier specifies types of special characters that DEC/Test Manager should ignore while performing the comparison. The /FULL qualifier

includes in the difference file a listing of text that was identical and the differences DEC/Test Manager encountered during the comparison. The /PARALLEL and /WIDTH qualifiers specify the format for the difference file.

After running and comparing a collection, you can review the test results and the comparison statuses with the DEC/Test Manager Review subsystem. The Review subsystem is a subsystem of DEC/Test Manager that gives you access to the results of a collection run and to other collection information. The Review subsystem is further described in Chapter 5.

4.4 Displaying the Collection Summary—SHOW COLLECTION

The SHOW COLLECTION command displays a summary of information for the collection. The format is

```
DTM SHOW COLLECTION collection-expression
```

The SHOW COLLECTION command takes the following qualifiers:

```
/BENCHMARK_DIRECTORY  
/BRIEF  
/CLASS  
/FULL  
/INTERMEDIATE  
/OUTPUT  
/TEMPLATE_DIRECTORY
```

The /BRIEF, /INTERMEDIATE, and /FULL qualifiers specify a brief, intermediate, or full form of collection information. The /OUTPUT[=file-spec] qualifier directs output to your screen (the default output device SYS\$OUTPUT), or to the specified file. If you do not specify a file, output is directed to the default output device. The /BENCHMARK_DIRECTORY and /TEMPLATE_DIRECTORY qualifiers display the current default benchmark and template directories for the collection. The /CLASS qualifier displays the DEC/CMS class, if any, associated with the collection benchmark and template files.

The following example displays a summary of information for the collection MARGINS_NOV. The default qualifier, /INTERMEDIATE, is in effect.

```

$ DTM SHOW COLLECTION MARGINS_NOV
Collections in DEC/Test Manager Library DRA1: [PROJECT.DTMLIB]

MARGINS_NOV    1 test    31-DEC-1985    11:09:17 "margin tests as of November"
               Command: CREATE COLLECTION MARGINS_NOV MARGINS/GROUP
                                   "margin tests as of November"

               Status: has been run, compared, not reviewed
               Successful count: 0      Unsuccessful count: 0
               New test count: 1       Updated test count: 0
               Test not run count: 0   Comparisons Aborted: 0
$

```

Note that with the /INTERMEDIATE qualifier, the collection summary displays the following:

- The collection name
- The number of tests in the collection
- The time the collection was created
- The command that created the collection and the remark associated with it
- The collection's status—whether it has been run, compared, reviewed, or rerun
- The status of the tests in the collection—how many are successful, unsuccessful, new, updated, not run, or whose comparison aborted

4.5 Deleting Collections—DELETE COLLECTION

You should delete a collection after you review it and no longer need the results. When a collection is deleted, all collection-related files are deleted. Benchmark files, test descriptions, and groups, including any groups created during a Review session, are not deleted.

To delete a collection, use the DELETE COLLECTION command. The format is

```
DTM DELETE COLLECTION collection-expression [remark]
```

The DELETE COLLECTION command takes the following qualifiers:

```

/[NO]CONFIRM
/[NO]LOG

```

As with all REMOVE and DELETE commands, the /CONFIRM qualifier is the default. This option displays each collection name before it is deleted and requests you to confirm that DEC/Test Manager should delete the collection.

The following example deletes the collection MARGINS_NOV. Note that the /CONFIRM qualifier is in effect.

```
‡ DTM DELETE COLLECTION MARGINS_NOV "no longer needed"
Confirm deletion of collection MARGINS_NOV [Y/N] (N): Y
‡DTM-S-DELETED, collection MARGINS_NOV deleted
‡
```

You can delete a collection before or after it executes. You cannot delete a collection while it is being run or while it is in use. After deleting a collection, you can reuse its name for another collection.

NOTE

Your ability to delete a collection depends on the protection of the files in the collection. If you encounter errors while attempting to delete a collection, see Chapter 2.

4.6 Recreating Collections—RECREATE

Use the RECREATE command to recreate an existing collection. You will want to recreate a collection if you have changed one or more of the files required by tests in the collection, or if you have changed other information in the library since you created the collection. Note that you cannot recreate a collection after you have deleted it.

The RECREATE command recreates a collection by first deleting the existing collection and all files related to it, except for the benchmark file. Then the collection is recreated using the CREATE COLLECTION command that originally created the collection with the latest version of all required files. The format is

```
DTM RECREATE collection-name [remark]
```

The collection-name parameter specifies the collection that is to be recreated.

The RECREATE command takes the /[NO]CONFIRM and /[NO]LOG qualifiers.

4.7 Stopping Collections—STOP

Use the STOP command to terminate a collection executing in batch. This command stops execution of the collection and cleans up the DEC/Test Manager library. Press CTRL/C (rather than CTRL/Y) to terminate a collection running interactively.

If you stop an executing collection other than with the STOP command, or if the system crashes while a collection is executing, errors will occur and you will not be able to review the collection. Pressing CTRL/C or typing the DTM STOP command to terminate a collection run allows DEC/Test Manager to restore its library to a consistent state and to perform necessary post-run cleanup after the collection run stops. See Chapter 5 for instructions on how to recover the library, review the executed tests, and rerun the tests that did not execute.

The format is

```
DTM STOP collection-name [remark]
```

The collection-name parameter specifies the collection that is to be stopped.

The STOP command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following example stops the collection MARGINS_NOV. Note that the /CONFIRM qualifier is in effect.

```
‡ DTM STOP MARGINS_NOV "Stopping collection run"  
Confirm stop of collection MARGINS_NOV [Y/N] (N): Y  
‡DTM-S-STOPPED, collection MARGINS_NOV stopped  
‡
```



Evaluating Test Results

This chapter presents an overview of the Review subsystem and tutorial material for using Review subsystem commands and operations. The following topics are discussed:

- Review subsystem terminology
- Invoking and using the Review subsystem
- Locating the results of a test run
- Displaying and printing output files
- Creating and replacing benchmark files
- Creating a group from the Review subsystem
- Reviewing partially run collections
- Leaving the Review subsystem
- Using the Review subsystem keypad

5.1 Overview of the Review Subsystem

The DEC/Test Manager Review subsystem allows you to examine the results generated by executing tests with DEC/Test Manager. As described in Chapter 4, result and difference files are created when you execute a collection. Using the Review subsystem, you can examine these files as well as the benchmark files for your tests. A collection must be executed and compared before it can be reviewed. See Section 5.7 for instructions for reviewing a partially run collection.

You can invoke the Analyzer of the VAX Performance and Coverage Analyzer from within the Review subsystem. The Analyzer allows you to examine performance and coverage data gathered by the Collector of the VAX Performance and Coverage Analyzer while DEC/Test Manager was running tests. Chapter 7 describes using DEC/Test Manager with the VAX Performance and Coverage Analyzer.

5.2 Review Subsystem Terminology

This section describes result descriptions, which summarize test output and comparison statuses and let you quickly evaluate tests.

5.2.1 Result Descriptions

For each test in a collection, DEC/Test Manager creates a **result description** that summarizes information about the test output files. A test's result-description-name is the same as its test-name. Each result description corresponds to a test description and contains the following information:

- The result-description-name.
- The comparison status of the test.
- Whether or not the output files exist. If the benchmark file exists, its file specification is displayed.

A result description has the following format:

```
Result Description (result-description-name)  Comparison Status : (successful,  
                                              unsuccessful, new test,  
                                              updated, not run,  
                                              or comparison aborted)
```

```
Benchmark File (file specification or does not exist)  
Result File (is present or does not exist)  
Difference File (is present or does not exist)
```

5.2.2 Output Files

After a collection has been executed and compared, each test is associated with one or more of the output files described in the following list. Depending on its comparison status, a test may have no output files associated with it, or it may have one or more output files associated with it. This is described further in Section 5.2.3. The output files include the following:

- The **benchmark file** for a test contains the expected output for the test. It is created using the Review subsystem UPDATE command. Benchmark file names can also be replaced and deleted with the MODIFY TEST_DESCRIPTION/BENCHMARK and MODIFY TEST_DESCRIPTION/NOBENCHMARK commands.

For an interactive test that has a benchmark file, DEC/Test Manager also creates the file test-name.BMK_SCREEN. This file contains printable copies of the screen images corresponding to the information in the benchmark file. This file is used only to print the screens in the benchmark file.

- The **difference file** for a test contains the differences between the benchmark and result files. It is created during comparison of the benchmark and result files. A difference file is created only if there are differences between the benchmark and result files.
- The **result file** for a test contains the results of a test's execution.

For an interactive test that has a result file, DEC/Test Manager also creates the file test-name.RES_SCREEN. This file contains printable copies of the screen images corresponding to the information in the result file. This file is used only to print the screens in the result file.

The Review subsystem PRINT and SHOW commands take the output file qualifiers, which allow you to specify that one or more output files are to be printed or displayed.

5.2.3 Comparison Status

For every test in a collection that runs to completion, DEC/Test Manager by default compares the result file with the benchmark file (if it exists) and reports the result as the **comparison status** for the test. The comparison status provides the following information:

- A **comparison aborted** status indicates that the test was run, but the comparison could not be completed. The benchmark file exists, but the result file and difference file might not.
- A **new test** status indicates that when the test was compared it had no benchmark file. A result file was produced, but no difference file exists.
- A **not run** status indicates that the test was part of a partially run collection and that the test did not run. A benchmark file might exist. No result file or difference file exists.
- A **successful** status indicates that when the test was compared it had a benchmark file and that the result file produced from this test execution matched the benchmark file. The result file has been deleted and no difference file exists.
- An **unsuccessful** status indicates that when the test was compared it had a benchmark file and that the result file produced from this test execution did not match the benchmark file. A difference file exists.
- An **updated** status indicates that the test has a benchmark file that was created from the result file since the test was executed in this collection. The benchmark file may have been created either in the current Review session or in a previous Review session. No result file or difference file exists.

Many Review subsystem commands take the comparison status qualifiers. This allows you to specify that the commands operate on sets of result descriptions with the specified comparison status. For more information, see Section 5.2.4.

5.2.4 Identifying Result Descriptions

When reviewing the results for a collection of tests, you should note the following general ideas.

- Each result description can be identified both by its result-description-name and by its comparison status.
- By specifying either result-description-expressions containing wild-cards or one or more comparison status qualifiers, or by specifying both, you can consider sets of result descriptions with similar characteristics.
- For the SHOW and PRINT commands, you can also specify output file qualifiers to print or display output files. For example, you can enter the SHOW * /SUCCESS/BENCHMARK command to display the benchmark files for all successful tests.
- When you enter a Review subsystem command that takes the result-description-expression and the comparison status qualifiers, the following information applies:
 - If you omit the result-description-expression and the comparison status qualifiers, DEC/Test Manager performs the operation on the current result description.
 - If you specify a result-description-name and omit the comparison status qualifiers, DEC/Test Manager performs the operation on the specified result description and makes it the current result description. For example, you can enter the SHOW/RESULT LMTEST5 command to display the result file for result description LMTEST5 and then make LMTEST5 the current result description.
 - If you specify several result descriptions by entering a result-description-expression and you omit the comparison status qualifiers, DEC/Test Manager performs the operation on all result descriptions matching the result-description-expression, but it does not change the current result description. For example, you can enter the SHOW /RESULT *LM* command to display the result files for all result descriptions matching *LM* without changing the current result description.
 - If you specify a single result-description-name and one or more comparison status qualifiers, DEC/Test Manager performs the operation on the specified result description, makes it the current result description, and ignores the comparison status qualifiers. For example, you can enter the PRINT/SUCCESS LMTEST2 command to print the result file for result description LMTEST2

and then make LMTEST2 the current result description. DEC/Test Manager ignores the /SUCCESS qualifier on this command and prints the result file by default.

- If you omit the result-description-expression parameter and include one or more comparison status qualifiers, DEC/Test Manager interprets this as the asterisk (*) wildcard parameter and operates on all result descriptions with the specified comparison status or statuses. For example, you can enter the SHOW/FILES/SUCCESS/NEW command to display the result files for all result descriptions.
- If you include both a result-description-expression and one or more comparison status qualifiers, DEC/Test Manager operates on all result descriptions that match both the result-description-expression and one of the comparison status qualifiers. For example, if you enter the INSERT/UNSUCCESS/NOT_RUN *LM* command, all test descriptions that match *LM* and have the unsuccessful or not run comparison status are marked for insertion into a group when you exit the Review subsystem.

5.3 Invoking and Using the Review Subsystem

The REVIEW command invokes the Review subsystem. The format is

```
DTM REVIEW collection-name
```

The REVIEW command takes the /READ_ONLY qualifier.

If you enter the REVIEW command without the /READ_ONLY qualifier, you are designated as the **primary reviewer** of the collection. Only one person at a time can be the primary reviewer. This reviewer can use all Review subsystem commands.

If you enter the REVIEW/READ_ONLY command, you are designated a **read-only reviewer** of the collection. You can browse through the result descriptions and print files, but you cannot make any changes to the result descriptions. Read-only reviewers cannot enter the UPDATE or INSERT commands. There can be multiple read-only reviewers for a collection.

If the primary reviewer is updating benchmark files when a read-only reviewer enters a SHOW command, the resulting display may be inaccurate. Under the same circumstances, the Collection Summary Information also may be incorrect and files queued for printing may disappear. As a read-only reviewer, you can avoid having files you select for printing disappear by using the PRINT/NOW command.

The following is a list of Review subsystem commands and how you can use them:

- BACK and NEXT, FIRST and LAST, and SELECT—to position you at a particular result description
- PRINT and SHOW—to view test results
- INSERT and UPDATE—to work with test results
- EXIT—to exit the Review subsystem and, with qualifiers, cancel commands issued during the Review session
- PCA—to invoke the Analyzer of the VAX Performance and Coverage Analyzer to examine performance and coverage data gathered by the Collector while your tests were running
- ATTACH and SPAWN—to create one or more subprocesses while you are in the Review subsystem and to switch control between them
- DEFINE/KEY—to define new keys on the Review subsystem keypad, or to redefine the default keypad definitions
- @file-spec—to execute a command file
- HELP—to display HELP information

5.4 Locating Test Results

Test results are accessed alphabetically by **result-description-name** and by comparison status. A test's result-description-name is the same as its test-name. When you enter the Review subsystem, you are not positioned at a result description. You can access a result description by entering a Review subsystem positioning command: FIRST, LAST, SELECT, NEXT, or BACK. By default, Review subsystem commands operate on the result description at which you are currently positioned, called the **current result description**.

The FIRST command moves you to the first result description in the collection; the LAST command moves you to the last result description.

The formats for these commands are

```
DTM_REVIEW> FIRST
```

```
DTM_REVIEW> LAST
```

The FIRST and LAST commands take no parameters or qualifiers.

The SELECT command moves you to a specific result description that you identify by its result-description-name. The format is

```
DTM_REVIEW> SELECT result-description-name
```

The SELECT command takes no qualifiers.

The NEXT command moves you forward and the BACK command moves you backward through the result descriptions in the collection. Using these commands you can move:

- To the next or previous result description
- To the next or previous result description with a particular comparison status
- Forward or backward a specified number of result descriptions
- Forward or backward a specified number of result descriptions with a particular comparison status

The formats for the NEXT and BACK commands are

```
DTM_REVIEW> NEXT [count]
```

```
DTM_REVIEW> BACK [count]
```

The optional count parameter indicates how many result descriptions you want to move forward or backward. The default is to move one result description in the specified direction.

The NEXT and BACK commands take the comparison status qualifiers alone or in combination with a count parameter value:

```
/COMPARISON_ABORTED  
/NEW  
/NOT_RUN  
/SUCCESSFUL  
/UNSUCCESSFUL  
/UPDATED
```

You can also move to the next result description by pressing RETURN.

5.5 Examining Test Results

The procedure for examining test results after executing tests is summarized in the following steps:

1. Enter the Review subsystem.
2. Examine the Collection Summary Information.
3. Examine the result description for each test.
4. Examine one or more of the output files referenced in the result description: the result, difference, and benchmark files.

Test results are examined with the SHOW and PRINT commands. The SHOW command displays information about result descriptions and displays their output files on your screen. The PRINT command prints copies of specified output files. Both commands allow you to specify a result-description-expression and one or more comparison status qualifiers. Thus, you can examine test results singly or in groups by similar result-description-names or by comparison status. Both commands also take the output file qualifiers. Thus, you can display and print output files singly or in groups.

5.5.1 Displaying Test Results

The SHOW command displays information about result descriptions and displays the output files associated with result descriptions.

The format is

```
DTM_REVIEW> SHOW [result-description-expression]
```

The result-description-expression identifies the result descriptions about which information is to be displayed. The default is to display information about the current result description.

The comparison status qualifiers allow you to examine groups of result descriptions with the same comparison status:

```
/COMPARISON_ABORTED  
/NEW  
/NOT_RUN  
/SUCCESSFUL  
/UNSUCCESSFUL  
/UPDATED
```

The output file qualifiers allow you to display specified output files singly or in groups:

```
/BENCHMARK  
/DIFFERENCE  
/RESULT
```

If you do not include an output file qualifier, DEC/Test Manager displays information about the specified result descriptions rather than an output file.

The SHOW command also takes the following qualifiers:

```
/FILES  
/OUTPUT  
/SUMMARY
```

The /FILES qualifier displays the comparison status for result descriptions and states whether each output file exists. The /FILES qualifier cannot be used with the output file and /SUMMARY qualifiers.

The /SUMMARY qualifier redisplay the Collection Summary Information—the information displayed when you invoke the Review subsystem. The /SUMMARY qualifier cannot be used with the comparison status, output file, and /FILES qualifiers.

5.5.1.1 Displaying Output Files for Noninteractive Tests

When you enter a SHOW command with an output file qualifier and specify a result description for a noninteractive test, DEC/Test Manager automatically recognizes the test as noninteractive and types the specified output file.

5.5.1.2 Displaying Output Files for Interactive Tests

When you enter a SHOW command with an output file qualifier and specify a result description for an interactive test, DEC/Test Manager automatically recognizes the test as interactive and displays the specified file screen by screen. The result and benchmark files are displayed in the same way. The difference file is displayed in a split-screen mode: one half of a result file screen is displayed on one half of the terminal screen, and one half of the corresponding benchmark screen is displayed on the other half of the terminal screen.

Displaying Result and Benchmark Files

When you enter a SHOW/RESULT or SHOW/BENCHMARK command, a banner screen (Screen 0) displays as in the following example:

Res
Screen 0

```
DEC Test Manager
INTERACTIVE DISPLAY
Version 2.0

Type PF2 for help.
Use keypad 0 to move to next screen.
Type CTRL/Z to exit.

Test          MAIL_TEST
Result File   DRA1:IPROJECT.DTMLIB.RNOCOLLECTMAIL_TE
```

ZK-4753-85

You can now do one of the following:

- Press PF2 to display HELP information.
- Press KP0 to display the first screen of the result or benchmark file.
- Press CTRL/Z to terminate the SHOW command and return control to the Review subsystem.

To move around in the result or benchmark file, you must press keypad keys. If your terminal does not have a keypad, the corresponding number keys give you the same movement, and pressing CTRL/H displays help. The keypad keys and their associated functions are shown in Table 5-1.

Table 5-1: Key Definitions for SHOW/RESULT and SHOW/BENCHMARK

Keypad Key	Key Definition
KP0	Displays the next screen.
KP1	Displays the previous screen.
KP5	Displays the first screen in the file.
KP7	Toggles the screen number display. If the screen number display is on the screen, pressing KP7 removes it. If the screen number display is not on the screen, pressing KP7 displays it.
PF2	Toggles the HELP display. If the HELP display is on the screen, pressing PF2 removes it. If the HELP display is not on the screen, pressing PF2 displays it.
CTRL/W	Refreshes the screen.
CTRL/Z	Terminates display of the result or benchmark file and returns control to the Review subsystem.

When you display a result or benchmark file, the specified file displays screen by screen. The screen number display, initially on the top right corner of the screen, shows the number of the screen and the type of file currently displayed.

You can move forward or backward screen by screen by pressing KP0 and KP1, or you can return to the first screen by pressing KP5. Pressing PF2 displays HELP information. Pressing CTRL/W refreshes the screen. Pressing CTRL/Z terminates display of the file and returns control to the Review subsystem.

Displaying Difference Files

When you enter a SHOW/DIFFERENCE command, a banner screen (Screen 0) displays as in the following example:

Interactive Compare	
Version 2.0	
Type PF2 for help.	
Test	MAIL_TEST
Result File	DRA1:[PROJECT.DTMLIB.RNOCOLLECT]MAIL_TE
Interactive Compare	
Version 2.0	
Type CTRL/Z to exit.	
Test	MAIL_TEST
Benchmark File	DRA1:[PROJECT.DTMLIB]MAIL_TEST.BMK;

Res	0
Screen	

Bmk	0
Screen	

ZK-4752-85

You can now do one of the following:

- Press PF2 to display HELP information.
- Press PF4 to select underlining rather than inverse bold for highlighting of differences. After a screen is displayed, you cannot change the way its differences are highlighted.
- Press KP0 to display the first screen of the difference file.
- Press CTRL/Z to terminate the SHOW command and return control to the Review subsystem.

To move around in the difference file, you must press keypad keys. If your terminal does not have a keypad, the corresponding number keys give you the same movement, and pressing CTRL/H displays HELP. The keypad keys and their associated functions are shown in Table 5-2.

Table 5-2: Key Definitions for SHOW/DIFFERENCE

Keypad Key	Key Definition
KP0	Displays the next screen.
KP1	Displays the previous screen.
KP3	Shifts split-screen mode. If the top (or bottom) half of the result and benchmark files is displayed, pressing KP3 displays the other half of the two screens. If the right (or left) half of the result and benchmark screens is displayed, pressing KP3 displays the other half of the two screens.
KP4	Changes highlighting of differences for screens that have not been displayed. By default, differences are highlighted in bold reverse video. This can be changed so that differences are underlined. Pressing KP4 changes highlighting for the screens that have not been displayed. After a screen is displayed, you cannot change the way its differences are highlighted.
KP5	Displays the first screen in the file.
KP7	Toggles the screen number display. If the screen number display is on the screen, pressing KP7 removes it. If the screen number display is not displayed, pressing KP7 displays it.
KP8	Toggles split-screen mode. If you are in full-screen mode, pressing KP8 puts you in split-screen mode. If you are in split-screen mode, pressing KP8 switches you between horizontal split-screen mode and vertical split-screen mode.
KP9	Toggles full-screen mode. If you are in split-screen mode, pressing KP9 puts you in full-screen mode. If you are in full-screen mode, pressing KP9 switches you between displaying full screens from the result and benchmark files.

Table 5-2 (Cont.): Key Definitions for SHOW/DIFFERENCE

Keypad Key	Key Definition
PF2	Toggles the HELP display. If HELP is displayed, pressing PF2 removes it. If HELP is not displayed, pressing PF2 displays it.
CTRL/W	Refreshes the screen.
CTRL/Z	Terminates display of the difference file and returns control to the Review subsystem.

When you display a difference file, by default you are in horizontal split-screen mode; the terminal screen is divided horizontally into two windows. A screen from the result file is displayed in the top window and the corresponding screen from the benchmark file is displayed in the bottom window. The screen number display, initially on the top right corner of each window, shows the number of the screen and the type of file currently displayed in that window.

By default, differences are highlighted in bold reverse video. Highlighting can be changed so that differences are underlined. After a screen is displayed, you cannot change the way its differences are highlighted. Pressing KP4 changes highlighting for all screens that you have not yet displayed. To change highlighting for the entire difference file, press KP4 when screen 0 displays, before viewing any other screens.

By default, you are in differences mode; only the screens that contain differences are displayed. You can move forward or backward to the next (or previous) screen containing differences by pressing KP0 (or KP1), or you can return to the first screen by pressing KP5. If you specified the /FULL qualifier on the COMPARE command, you will not be in differences mode; all screens will be displayed, not just the screens containing differences.

Pressing KP8 when you are in split-screen mode switches you between horizontal and vertical split-screen mode. By default, you are in horizontal split-screen mode; pressing KP8 divides the terminal screen vertically into two windows. The left half of the result screen is displayed in the left window and the left half of the benchmark screen is displayed in the right window. The screen number display is in the top right corner of each window display. When you are in full-screen mode, pressing KP8 returns you to split-screen mode.

Press KP3 to shift the split screen. By default, the terminal screen is divided horizontally into two windows with the top half of the result screen displayed in the top window and the top half of the benchmark screen displayed in the bottom window. Pressing KP3 displays the bottom half of each screen in its window. When the terminal screen is divided vertically into two windows, pressing KP3 reverses the right and left halves of the screens currently displayed.

When you are in split-screen mode, press KP9 to change to full-screen mode. Pressing KP9 while you are in full-screen mode changes between displaying full screens from the result file and displaying full screens from the benchmark file.

Pressing PF2 displays HELP information. Pressing CTRL/W refreshes the screen. Pressing CTRL/Z terminates the display of the file and returns control to the Review subsystem.

5.5.2 Printing Test Results

The PRINT command marks the specified output files for placement in the default printer queue SYS\$PRINT. By default, the selected output files are concatenated and released for printing when you leave the Review subsystem with the EXIT command or by pressing CTRL/Z. If you leave the Review subsystem by typing the EXIT/NOPRINT command or by pressing CTRL/C, the marked files are not printed.

The format is

```
DTM_REVIEW> PRINT [result-description-expression]
```

The result-description-expression identifies the result descriptions whose output files are to be marked for printing. The default is to mark output files associated with the current result description.

The comparison status qualifiers allow you to print output files for result descriptions with the specified comparison status:

```
/COMPARISON_ABORTED  
/NEW  
/NOT_RUN  
/SUCCESSFUL  
/UNSUCCESSFUL  
/UPDATED
```

The output file qualifiers allow you to print specified output files singly or in groups:

```
/BENCHMARK  
/DIFFERENCE  
/RESULT
```

If you do not include an output file qualifier, DEC/Test Manager marks for printing the result file associated with the current result description.

The PRINT command also takes the following qualifiers:

```
/[NO]LOG  
/NOW  
/SELECTED
```

The /**[NO]LOG** qualifier controls whether DEC/Test Manager displays informational and success messages on your screen.

The /**NOW** qualifier concatenates all files specified on the current PRINT command and immediately places them in the print queue.

The /**SELECTED** qualifier concatenates all files already marked for printing with the currently specified files and immediately places them in the print queue.

If you select a result file for printing and subsequently update the benchmark file for that test before the result file has been queued for printing, the result file is deleted and is no longer printable.

5.6 Working with Test Results

The procedures for working with test results from within the Review subsystem are summarized in the following section. The UPDATE command creates benchmark files for new tests and updates existing benchmark files. The INSERT command marks test descriptions for inclusion into a group to facilitate retesting.

5.6.1 Creating Benchmark Files from Within the Review Subsystem

While in the Review subsystem, you can create or replace a benchmark file for a test with the UPDATE command. The UPDATE command makes the current result file the new benchmark file for the specified result descriptions. If a benchmark file currently exists in the DEC/Test Manager library, it is deleted. If a benchmark file currently exists but it is outside the DEC/Test Manager library, DEC/Test Manager informs you of this and replaces it as the benchmark file, but does not delete the old file.

The format is

```
DTM_REVIEW> UPDATE [result-description-expression]
```

The result-description-expression identifies the result descriptions whose benchmark files are to be updated. The default is to update the benchmark files for the current result description.

The UPDATE command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The `/[NO]CONFIRM` qualifier determines whether DEC/Test Manager prompts you before updating each benchmark file. The `/[NO]LOG` qualifier determines whether DEC/Test Manager displays informational and success messages on your screen.

The example in Section 5.6.1.1 shows how to use Review subsystem commands to replace a test's benchmark file. The example in Section 5.6.1.2 shows how you can use Review subsystem commands to create a benchmark file for a new test.

5.6.1.1 Updating an Existing Benchmark File

You might want to update an existing benchmark file if you changed the software being tested in a way that would change the expected results for a test. The following examples assume that you are in the Review subsystem and that you have already examined the result file or difference file and determined that the result file should replace the old benchmark file.

First, locate the result description for the test by typing the SELECT command:

```
DTM_REVIEW> SELECT LMTTEST1
```


Then, type the UPDATE command:

```
DTM_REVIEW> UPDATE
```

The result file is now renamed as the benchmark file and the reference to the old benchmark file is removed. If the old benchmark file is in the DEC/Test Manager library, it is deleted.

When you update an existing benchmark file that is stored in a DEC/CMS library, DEC/Test Manager reserves the existing benchmark element in the DEC/CMS library and replaces it with the result file from the current test run. If you specify a DEC/CMS generation for the existing benchmark file with the CREATE COLLECTION/CLASS command, DEC/Test Manager inserts the updated benchmark file as a new generation in the specified DEC/CMS class.

If you are updating the benchmark file for an interactive test, the file containing printable benchmark screens is also updated.

5.6.1.2 Creating a Benchmark File for a New Test

This section describes how to use the Review subsystem to create a benchmark file for a new test. The examples show how to (1) select a collection for reviewing, (2) locate the result description for a test, (3) display a result file, and (4) create a benchmark file for the test by renaming the result file as the benchmark file.

After a collection has been executed and compared, you can review it using the DTM REVIEW command. This command places you in the Review subsystem and displays the Collection Summary Information. To review a collection named LMCOLLECT1, type the following:

```
‡ DTM REVIEW LMCOLLECT1
Collection LMCOLLECT1 with 1 test was created on 31-OCT-1985 07:19:16
by the command:
  CREATE COLLECTION LMCOLLECT1 LMTEST1 "run of simple left margin test"
  Last Review Date = not previously reviewed
  Success count = 0
  Unsuccessful count = 0
  New test count = 1
  Updated test count = 0
  Comparisons aborted = 0
  Test not run count = 0
DTM_REVIEW>
```

The collection summary displays the following information:

- Collection name
- Number of tests in the collection

- Date and time the collection was created
- Command that created the collection
- Whether the collection was reviewed previously and, if so, when
- A summary of the comparison status for all tests in the collection

You can see that the collection was not reviewed previously and that it contains one new test, LMTEST1. To access the result description for the test LMTEST1, type the following:

```
DTM_REVIEW> SELECT lmtest1
Result Description LMTEST1          Comparison status : New test
```

The result description tells you that this is a new test. A new test is one for which a benchmark file does not exist. As a next step, you may want to examine the result file to determine whether the results are correct. If they are correct, you can then make this file the benchmark file for the test.

Use the SHOW command to display the result file for LMTEST1, or use the PRINT command to queue the result file to the default printer.

Enter the following command to display the result file for LMTEST1. It is unnecessary to specify the result-description-name, because LMTEST1 is the current result description.

```
DTM_REVIEW> SHOW/RESULT
```

Because LMTEST1 is a noninteractive test, DEC/Test Manager types the result file. If the result file contains the expected results, use the UPDATE command to make this result file the benchmark file. To do this, type the following:

```
DTM_REVIEW> UPDATE
```

This command creates a benchmark file for LMTEST1 from its result file, deletes the result file, and changes the comparison status of the test from new to updated. Enter the SHOW/SUMMARY command to verify this.

```
$ DTM_REVIEW> SHOW/SUMMARY
Collection LMCOLLECT1 with 1 test was created on 31-OCT-1985 07:19:16
by the command:
    CREATE COLLECTION LMCOLLECT1 LMTEST1 "run of simple left margin test"
    Last Review Date = 02-NOV-1985 08:19:20
    Success count = 0
    Unsuccessful count = 0
    New test count = 0
    Updated test count = 1
    Comparisons aborted = 0
    Test not run count = 0
DTM_REVIEW>
```

DEC/Test Manager stores the benchmark file in the DEC/Test Manager library and uses it as the benchmark file for LMTEST1 when the test is run in future collections. When you create a new benchmark file in a DEC/CMS library, DEC/Test Manager creates a new benchmark element in the specified DEC/CMS library. If you specify a DEC/CMS class with the CREATE COLLECTION/CLASS command, DEC/Test Manager inserts the new benchmark file as the first generation in the class.

If you are creating the benchmark file for an interactive test, the file containing printable benchmark screens (test-name.BMK_SCREEN) is also created.

To print the newly created benchmark file, type the following:

```
DTM_REVIEW> PRINT/BENCHMARK
```

This marks the file to be sent to the default printer queue when you end the Review session.

Note that when you omit the result-description-expression, the SHOW, UPDATE, and PRINT commands all refer to the current result description unless you include a comparison status qualifier. The benchmark file is updated when you enter the command. By default, files marked for printing are printed when you leave the Review subsystem. To leave the Review subsystem, type the following:

```
DTM_REVIEW> EXIT
```

5.6.2 Creating a Group from Within the Review Subsystem

While you are reviewing a collection, you can insert tests into a single new group. DEC/Test Manager creates the group when you exit the Review subsystem. Use the EXIT/NOINSERT command if you do not want a group to be created from the marked tests. Any group you create becomes part of the DEC/Test Manager library and has the same properties as any group you create using the CREATE GROUP command. The CREATE GROUP command, as well as other information about DEC/Test Manager groups, is described in Chapter 6.

You can, for example, create a group that contains all the tests in a collection whose comparison status was unsuccessful. This is useful if you want to rerun only those unsuccessful tests.

Only one group can be created from a collection during a Review session. To create more than one group from a collection, mark the tests to be inserted into the first group, exit from the Review subsystem to create that group, then initiate another Review session for the same collection and create the second group.

Use the INSERT command to group the test descriptions. The INSERT command marks the test description corresponding to the specified result descriptions for inclusion in a group. By default, DEC/Test Manager creates the group when you exit from the Review session. If you exit the Review subsystem with the EXIT/NOINSERT command or by pressing CTRL/C, the group is not created.

The format is

```
DTM_REVIEW> INSERT [result-description-expression]
```

The result-description-expression identifies the test descriptions for insertion into the group. The default is to insert into the group the test description associated with the current result description.

The comparison status qualifiers allow you to include sets of test descriptions with the same comparison status:

```
/COMPARISON_ABORTED  
/NEW  
/NOT_RUN  
/SUCCESSFUL  
/UNSUCCESSFUL  
/UPDATED
```

The INSERT command also takes the /[NO]CONFIRM qualifier which determines whether DEC/Test Manager prompts for confirmation before marking each test description for insertion.

DEC/Test Manager automatically names the group you create from within the Review subsystem. Use this name to access the group. The group name has this format:

```
collection-name$DTM_#
```

The pound sign (#) is a number supplied by DEC/Test Manager which differentiates the groups created in different Review sessions.

For example, if the following command is issued while the collection ALLTESTS is being reviewed, DEC/Test Manager creates a group that contains all the tests in the collection whose comparisons were unsuccessful.

```
DTM_REVIEW> INSERT /UNSUCCESSFUL
```

When DEC/Test Manager creates the group, it uses ALLTESTS as the collection-name. If no other group names begin with ALLTESTS\$DTM_, DEC/Test Manager assigns 1 to the # field. The group name is ALLTESTS\$DTM_1.

If you then create a new collection, SMITHTESTS, and run it while reviewing the collection SMITHTESTS, you create a group of unsuccessful tests. This group is named SMITHTESTS\$DTM_1. Later, you decide you no longer need the collection and delete it.

Next, you make fixes to the software tested by the unsuccessful tests and decide to run these tests again to check your fixes. You create another collection and give it the same name, SMITHTESTS. While reviewing the second collection, you again group the unsuccessful tests. The group of unsuccessful tests you create while reviewing the second collection is named SMITHTESTS\$DTM_2, even though this is the first time this particular collection has been reviewed.

Note that the name for a group created from within the Review subsystem with the INSERT command is based on the name of the collection being reviewed when the group is created. Also, a collection-name can be used again after a previous collection of that name has been deleted.

5.7 Reviewing Partially Run Collections

A partially run collection may occur for one of the following reasons:

- You may have stopped a collection by typing the DTM STOP command, if it was executing in batch, or by pressing CTRL/C, if it was running interactively.
- The system may have crashed while your collection was executing.
- You may have stopped a collection of tests running in batch with the DCL DELETE/ENTRY command.
- You may have pressed CTRL/Y to stop a collection that was running interactively.
- Someone may have stopped the process executing the collection.

The DTM STOP command is the recommended way to terminate a collection executing in batch. This command cleans up the DEC/Test Manager library after stopping the collection. Using this command, you must compare the partially run collection (with the DTM COMPARE command) before reviewing it. Pressing CTRL/C is the recommended way to stop a collection running interactively.

If you stop a collection other than by entering the STOP command or by pressing CTRL/C, errors may occur and you will not be able to compare or review the collection. Before using the collection, you must enter the VERIFY/RECOVER command to clean up the library, correct the errors, and mark the tests that did not run. Then, you must enter the COMPARE command to compare the partially run collection. Finally, you must enter the REVIEW command to initiate a Review session for the partially run collection.

In both cases, all tests that did not run are marked with the not run comparison status. If the collection is stopped after at least some of the tests have been compared, those tests that originally compared successfully will now be marked not run since DEC/Test Manager deleted the result file after the original successful comparison.

Reviewing a partially run collection is especially important if the collection is large. Following the instructions just described, you should prepare the partially run collection for review. Then, from the Review subsystem, examine the tests that ran and use the INSERT/NOT_RUN command to create a group containing all the tests which did not run. This group can then be included in a collection, executed, and reviewed. Creating a group while reviewing a collection is described in Section 5.6.2.

5.8 Leaving the Review Subsystem

The EXIT command terminates the Review session.

The format is

```
DTM_REVIEW> EXIT
```

The EXIT command takes the following qualifiers:

```
/[NO]INSERT  
/[NO]PRINT
```

By default, the /INSERT and /PRINT qualifiers are associated with the EXIT command. DEC/Test Manager concatenates and prints files marked for printing and inserts marked test descriptions into a group. Pressing CTRL/Z to terminate a Review session has the same effect as entering the EXIT/PRINT/INSERT command.

Enter the EXIT/NOPRINT/NOINSERT command to terminate the Review session without printing the marked files or creating a group. Pressing CTRL/C has the same effect as entering the EXIT/NOPRINT/NOINSERT command.

5.9 Canceling Review Subsystem Commands

Press CTRL/C to cancel a transaction while it is being processed and to return control to the DTM_REVIEW> prompt. If you press CTRL/C during a wildcard transaction that updates the library, DEC/Test Manager finishes the current transaction, but does not continue. If you press CTRL/C at the DTM_REVIEW> prompt, DEC/Test Manager terminates the Review session as if you had entered the EXIT/NOPRINT/NOINSERT command.

5.10 Using the Review Subsystem Keypad

You can define the keys on your terminal keypad to execute Review subsystem commands whenever you are in the DEC/Test Manager Review subsystem. Thus, you can execute a Review subsystem command by pressing only one or two keypad keys.

5.10.1 Default Key Definitions

On the terminal keypad, most of the default key definitions execute one or two Review subsystem commands. Several keys are undefined. The Review subsystem default key definitions are shown in Figure 5-1.

Figure 5-1: Review Subsystem Default Key Definitions

REVIEW
GOLD_REVIEW

PF1	PF2	PF3	PF4
GOLD	HELP KEYPAD (HELP)		
7 SHO/DIF PRI/DIF	8 SHO/RES PRI/RES	9 SHO/BEN PRI/BEN	— SHO/SUM
4 PRI/NOW	5	6 UPDATE	' SPAWN ATT/PAR
1 NEXT/UN BACK/UN	2 NEXT/NE BACK/NE	3 NEXT/UP BACK/UP	ENTER
0 NEXT BACK	• FIRST LAST		

The RETURN key and the ENTER key move you to the next result description.

ZK-4749-85

To execute one of the commands associated with a keypad key (the top function listed on each key and marked REVIEW in Figure 5-1), press the designated keypad key. For example, press KP1 to execute the NEXT/UNSUCCESSFUL command and to move forward to the next unsuccessful test.

To execute the other command associated with the keypad key, (the bottom function listed on each key and marked GOLD_REVIEW in Figure 5-1), press PF1 (the GOLD key) followed by the designated keypad key. For example, press PF1 (the GOLD key) followed by KP1 to execute the BACK/UNSUCCESSFUL command and to move back to the previous unsuccessful test.

5.10.2 Redefining the Key Definitions

You can redefine the Review subsystem commands associated by default with the keypad keys. Using the DEFINE/KEY command, you can replace any or all of the keypad key definitions, you can reorganize the key definitions, or you can define the keypad keys that do not have default definitions. For example, KP4 does not have a default definition. To define KP4 to invoke the Analyzer of the VAX Performance and Coverage Analyzer, enter the following command:

```
DTM_REVIEW> DEFINE/KEY KP4 "PCA" /TERMINATE
```

After you define KP4 to execute this command, press KP4 to invoke the Analyzer of the VAX Performance and Coverage Analyzer.



Enhancing the Test System

This chapter describes the following optional test system enhancements that allow you to tailor DEC/Test Manager to your specific testing needs:

- Prologue and epilogue files
- Variables
- Groups
- Keypads
- Correcting errors in the library
- Command files
- Initialization files
- Spawning subprocesses
- Filters
- History logging

6.1 Using Prologues and Epilogues

Prologues and epilogues allow you to control the environment in which DEC/Test Manager runs your tests. Prologues and epilogues are command files that you create and associate with tests or collections. They can be used as setup, cleanup, or filter files for tests or collections. You must store prologues and epilogues outside the DEC/Test Manager library in directories or in DEC/CMS libraries.

The various prologue and epilogue files are defined and used as follows:

- The **collection prologue** file, a prologue file associated with a collection, is executed whenever the collection is executed after the global variables and `DTM$COLLECTION_NAME` are defined, but before any tests are executed.
- The **test prologue** file, a prologue file associated with a specific test description, is executed whenever the test is executed and before the test template file is executed.
- The **test epilogue** file is executed whenever the test is executed after the test template executes and `DTM$RESULT` is defined, but before the DEC/Test Manager-provided filters are run.
- The **collection epilogue** file is executed whenever the collection is executed. It is the last file executed.

For more information about how DEC/Test Manager executes collections and tests and their associated prologue and epilogue files, see Section 4.2 and Figure 4-1.

6.1.1 Test Prologues and Epilogues

Test prologue and epilogue files are associated with a specific test description. They are executed whenever the test is executed, as described in Section 6.1. A test prologue file is named in the prologue field of a test description. It can set up an environment for the test template file. For example, a test prologue file can define local variables, `FETCH` elements from a DEC/CMS library, or set default values. A test epilogue file is named in the epilogue field of a test description. It can clean up the environment after the template file is run. Test epilogue files can also remove run-dependent information, such as the date, from a test's result file. You can usually determine the need for such a file by running a test once and checking for run-dependent information in the result file. For more information, see Section 6.3.1.

Use the `CREATE TEST_DESCRIPTION` or `MODIFY TEST_DESCRIPTION` commands with the `/PROLOGUE` and `/EPILOGUE` qualifiers to associate prologue and epilogue files with a test description, or to replace the prologue or epilogue file associated with a test description.

Use the `MODIFY TEST_DESCRIPTION` command with the `/NOPROLOGUE` and `/NOEPILOGUE` qualifiers to remove prologue and epilogue files from a test description without replacing them. This does not delete the file; it dissociates the prologue or epilogue from the test description.

6.1.2 Collection Prologues and Epilogues

Collection prologue and epilogue files are associated with collections. They are executed whenever the collection is executed, as described in Section 6.1. You can create a default collection prologue or epilogue file by using the `SET PROLOGUE` and the `SET EPILOGUE` commands. All subsequently created collections invoke these default prologue and epilogue files whenever they are executed, if you do not explicitly specify a different collection prologue or epilogue file. Use the `CREATE COLLECTION` command and the `/PROLOGUE` or `/EPILOGUE` qualifier to specify a prologue or epilogue file other than the default for a specific collection. Override the default prologue or epilogue file for a collection, without replacing it, by using the `/NOPROLOGUE` or `/NOEPILOGUE` qualifier on the `CREATE COLLECTION` command. Cancel the default collection prologue or epilogue file by using the `SET NOPROLOGUE` or `SET NOEPILOGUE` command.

6.2 Using Variables

In DEC/Test Manager, a **variable** is a user-defined VAX/VMS symbol or logical name. DEC/Test Manager stores variables and uses them when executing tests. Variables can be referred to in template, prologue, and epilogue files. Variables provide a convenient way for you to tailor a single template, prologue, or epilogue file so that it can be used with multiple tests.

A variable can be global or local in scope. A **global variable** is accessible to any template, prologue, or epilogue file in a collection. All global variables in the current DEC/Test Manager library are defined at the beginning of every collection run, whether or not the variable is used in the collection. In contrast, a **local variable** is defined only while the test with which it is associated is running. Local variables can be used by the prologue, template, and epilogue files associated with this test.

6.2.1 Creating Variables—CREATE VARIABLE

The CREATE VARIABLE command adds variables one at a time to the DEC/Test Manager library. The format is

```
DTM CREATE VARIABLE variable-name variable-value [remark]
```

The variable-name and variable-value parameters specify the variable's name and its value. Note that you cannot use the variable names P1 through P8, nor can you use variable names beginning with the prefix DTM\$. These variable names are reserved exclusively for use by DEC/Test Manager. You receive a warning if you attempt to create a variable of this form.

The CREATE VARIABLE command takes the following qualifiers:

```
/GLOBAL  
/LOCAL  
/[NO]LOG  
/LOGICAL  
/NUMERIC  
/STRING  
/SYMBOL
```

The /LOGICAL or /SYMBOL qualifier defines the variable's use as a logical name or as a symbol. The default usage is as a symbol. The /NUMERIC and /STRING qualifiers further define a symbol as a numeric value or as a string. The default is numeric value for an unquoted value and string for a quoted value. The /GLOBAL or /LOCAL qualifier defines the variable's scope as global or local. The default scope is local.

In using the /NUMERIC and /STRING qualifiers with variables that are symbols, DCL operators such as the plus sign (+) and minus (-) must be placed in a quoted string to be used as arithmetic operators. DEC/Test Manager attempts to prevent the generation of syntactically incorrect DCL assignment statements. Therefore, numeric symbol variables and logical variables cannot have a null value.

After you have added a variable to the DEC/Test Manager library, you can use it in a test's template, prologue, or epilogue file. If a variable is global, it is available to all template, prologue, and epilogue files in a collection at the time the collection is run. Section 6.2.6 describes how to override the definition of a global variable for a particular test. If the variable is local, you must associate it with a test description before the prologue, epilogue, or template associated with that test description can reference the variable.

The following is an example of using the CREATE VARIABLE command. If you have a test that issues many SUBMIT commands and you do not want to print all the LOG files that the test generates, you can create a variable with variable-name SUBMIT and give it the variable-value SUBMIT/NOPRINTER.

```
⌘ CREATE VARIABLE/GLOBAL/SYMBOL SUBMIT "SUBMIT/NOPRINTER"  
_Remark: "Redefine the SUBMIT command"
```

Any test using the SUBMIT command now uses the new definition, as if you had entered the following command before executing the test:

```
⌘ SUBMIT ::= SUBMIT/NOPRINTER
```

6.2.2 Modifying Existing Variables—MODIFY VARIABLE

Use the MODIFY VARIABLE command to modify one or more variable characteristics, for example, to change the default value of a global variable. The format is

```
MODIFY VARIABLE variable-expression [remark]
```

The variable-expression parameter can be a variable-name, a wildcard character, a wildcard in combination with a variable-name, or a list of these separated by commas.

The MODIFY VARIABLE command takes the following qualifiers:

```
/GLOBAL  
/LOCAL  
/[NO]LOG  
/LOGICAL  
/NUMERIC  
/[NO]REMARK=quoted-string  
/STRING  
/SYMBOL  
/VALUE=variable-value
```

The /GLOBAL and /LOCAL qualifiers change the variable's scope. The /LOGICAL and /SYMBOL qualifiers change the variable's use. The /NUMERIC and /STRING qualifiers change the type of a variable used as a symbol. The /[NO]REMARK qualifier removes or replaces the variable's remark. The /VALUE qualifier replaces the variable's value.

6.2.3 Deleting Variables—DELETE VARIABLE

To delete a variable from the DEC/Test Manager library, use the DELETE VARIABLE command. The format is

```
DTM DELETE VARIABLE variable-expression [remark]
```

The variable-expression parameter can be a variable-name, a wildcard character, a wildcard in combination with a variable-name, or a list of these separated by commas.

The DELETE VARIABLE command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

DEC/Test Manager will not delete a variable that is associated with a test description. If you attempt to delete several variables with a variable-expression and one or more of them is associated with a test description, DEC/Test Manager deletes only those variables not associated with a test description. Use the MODIFY TEST_DESCRIPTION/NOVARIABLE command to dissociate variables from test descriptions.

6.2.4 Using Global Variables

Global variables can be accessed by any test in a collection if the test includes the variables in its template, prologue, or epilogue file. You do not need to associate a global variable with individual test descriptions. When a collection is created, DEC/Test Manager associates all existing global variables with it and defines them at the start of every collection run. Do not assign values to global variables from within template, prologue, or epilogue files. Instead, let DEC/Test Manager assign the indicated value before the template, prologue, or epilogue file is executed. If you redefine a global variable or create a new global variable in the DEC/Test Manager library, use the RECREATE command to recreate the collection and associate it with the new global variables.

To use a global variable, do the following:

- Define the variable as global with the CREATE or MODIFY VARIABLE command.
- Use the global variable in the template, prologue, and epilogue files of specific test descriptions.

Global variable values can be overridden for an individual test that requires special handling. See Section 6.2.6 for more information.

6.2.5 Using Local Variables

A local variable is accessible to an individual test if it is referenced in its test description.

To use a local variable, do the following:

- Define the variable as local with the CREATE or MODIFY VARIABLE command.
- Use the variable in the template, prologue, or epilogue file of specific test descriptions.
- Associate the variable with one or more test descriptions.
 - To associate local variables with a new test description, use the CREATE TEST_DESCRIPTION/VARIABLE command.
 - To associate local variables with an existing test description, use the MODIFY TEST_DESCRIPTION/VARIABLE command.

To dissociate variables from an existing test description, use the MODIFY TEST_DESCRIPTION/NOVARIABLE command.

6.2.6 Overriding Variable Default Values

Most tests use a variable's default value. However, certain tests may require special handling and, thus, require special variable values. For example, you may want to use one template file to run several tests. Do this by using a variable in the template file and by overriding the variable's value for each test description.

You can override a variable's default value for any number of test descriptions, but doing so affects all users of that DEC/Test Manager library. To override a variable's default value for a particular test, associate the new value with its test description:

- When creating a test description, use the CREATE TEST_DESCRIPTION/VARIABLE command.
- If the test description exists, use the MODIFY TEST_DESCRIPTION/VARIABLE command.

In the following example, the first command creates the variable `TEMPLDIR` and assigns it a value of `DRA1:[PROJECT.DSRLIB]`. The second command modifies the variable's value so that when it is used with the test description `LMTEST1`, its value is `DRA1:[MY.DSRLIB]`.

```
⌘ DTM CREATE VARIABLE TEMPLDIR DRA1:[PROJECT.DSRLIB]/LOGICAL /GLOBAL "TEMPLATE DIRECTORY"  
%DTM-S-CREATED, logical variable TEMPLDIR created  
⌘ DTM MODIFY TEST_DESCRIPTION LMTEST1/VARIABLE=TEMPLDIR=DRA1:[MY.DSRLIB]  
_Remark: "Change variable value when used in LMTEST1"  
%DTM-S-MODIFIED, test description LMTEST1 modified  
⌘
```

You can override the default value of a global variable when you create a collection. To do this, use the `CREATE COLLECTION/VARIABLE` command. The following example creates the collection `DSRTESTS` and changes the value of `TEMPLDIR` to `DRA1:[SMITH.DSRLIB]` for this collection only.

```
⌘ DTM CREATE COLLECTION DSRTESTS * /VARIABLE=TEMPLDIR="DRA1:[SMITH.DSRLIB]"--  
_⌘ "New template directory for this collection"
```

Note that test descriptions in collection `DSRTESTS` that are explicitly associated with the variable `TEMPLDIR` are not permanently affected by the override value. For example, `LMTEST1` in `DSRTESTS` will still have the value `DRA1:[MY.DSRLIB]` despite the collection override value.

You should override a variable with care. Because you are changing the variable's value from earlier test runs, the actual test output may not match the expected test output. In addition, any changes in variable values may affect the prologue and epilogue files. As a result, you must examine the differences and result files to discover whether the actual test output is what you expected.

The following example shows the creation of a general template file using a variable. Template file `LMTEMPLATE.COM` invokes `DSR` specifically to run the input file `LEFTMARGIN1.COM`.

```

$ ! TEMPLATE for a simple test of the DSR left margin (.LM) command
$ !
$ ! Display the input file
$ !
$ TYPE DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ ! Run DSR on the left margin test
$ !
$ DSR DRA1:[PROJECT.RNO]LEFTMARGIN1.RNO
$ !
$ ! Show the output of this runoff test
$ !
$ TYPE LEFTMARGIN1.MEM
$ !
$ ! Cleanup
$ !
$ DELETE LEFTMARGIN1.MEM;1

```

Presumably, all other template files for this project have a similar structure, in which the following is true:

- The DSR input file is typed to SYS\$OUTPUT.
- DSR is run on this input file.
- The output file (MEM) is typed to SYS\$OUTPUT.

However, it is simpler to create a single template file, TEMPLATE.COM, that runs DSR over a general DSR input file:

```

$ ! TEMPLATE for a simple test of the DSR command
$ !
$ ! Display the input file
$ !
$
$ TYPE TEMPLDIR:'inputfile'.RNO
$
$ !
$ ! Run DSR
$ !
$
$ DSR TEMPLDIR:'inputfile'.RNO
$
$ !
$ ! Show the output of DSR
$ !
$
$ TYPE 'inputfile'.MEM
$ !
$ DELETE 'inputfile'.MEM;1

```

To use the more general template file, you must identify to DEC/Test Manager the VAX/VMS symbol (INPUTFILE) named in that template file. To do this, create a DEC/Test Manager local variable called INPUTFILE and specify that it is a VAX/VMS string symbol to be used locally for each test. The value here is "", and a remark is given to further identify the variable.

```

$ DTM CREATE VARIABLE inputfile "" /SYMBOL/STRING/LOCAL-
_.$ "input file symbol for file template.com"
%DTM-S-CREATED, symbol variable INPUTFILE created

```

Now that the variable is available in the DEC/Test Manager library, associate the variable with each test description that uses the template file TEMPLATE.COM. When you associate the local variable INPUTFILE with each test description, you must specify the variable's value for that test because the variable has a default value of "". To omit the variable's value would cause the template file to run DSR on file "".RNO, which does not exist.

Modify the existing test description LMTEST1 to use this new template file and variable. Note that the resulting test description is exactly equivalent to the original.

```

$ DTM MODIFY TEST lmtest1 /TEMPLATE=templib:template.com /VAR=inputfile=leftmargin1
_Remark: "Associating new template and variable with LMTEST1"
%DTM-S-MODIFIED, test description LMTEST1 modified

```

```

$ DTM SHOW TEST lmtest1/FULL

```

Test descriptions in DEC/Test Manager Library DISK\$USER01:[PROJECT.DTMLIB]

```

LMTEST1      "left margin test"
  Template   =  TEMPLIB:TEMPLATE.COM
  Benchmark  =  LMTEST1.BMK
  Prologue   =  None Specified
  Epilogue   =  None Specified
  Variables  =  INPUTFILE = "LEFTMARGIN1"
  Groups     =  None Specified
  Type       =  Non-Interactive
  Filters    =  None Specified
  Comparison type = no local override

```

```

$

```

Another test description may use this same template file if you assign the variable INPUTFILE a different value, for example:

```

$ DTM CREATE TEST  pgtest3      "complex test of paging command" -
_.$ /TEMPLATE=templib:template.com /VAR=inputfile=pagesource1 -
_.$ /BENCHMARK=pagesource3.bmk
%DTM-S-CREATED, test description PGTEST3 created

```

6.3 Variables Defined by DEC/Test Manager

DEC/Test Manager supplies several built-in variables that can be used in template, prologue, and epilogue files. This section describes them and shows an example of how to use each one. For more information about the way DEC/Test Manager uses these variables when executing collections, see Section 4.2 and Figure 4-1.

6.3.1 DTM\$RESULT Logical Name

DEC/Test Manager establishes the VAX/VMS logical name DTM\$RESULT as the logical equivalent to the file specification for the test result file. This logical name exists only while the test epilogue file executes. DTM\$RESULT is defined immediately after the test template file executes and just before the test epilogue file executes. It is deassigned after the test epilogue file executes.

DTM\$RESULT allows you to create the epilogue file to filter run-dependent information from the result file. To do this, the epilogue file runs the result file through a text editor, such as EDT.

The following example shows an epilogue file that invokes EDT to remove all lines from the result file that contain VAX/VMS run information on the amount of memory used. The epilogue file deletes all lines containing "Memory Used:".

```
#! MEM.FIL -- Eliminate any "Memory Used:"
#!          messages from .RES files.
$ EDIT/EDT DTM$RESULT
c;32767('Memory Used:' dl) ex
EXIT
$ PURGE DTM$RESULT
```

6.3.2 DTM\$TEST_NAME Local String Symbol

DEC/Test Manager establishes the VAX/VMS local string symbol DTM\$TEST_NAME as the test-name field of the test description. DTM\$TEST_NAME can be used in template files, in test epilogue files, and in test prologue files.

The following example shows how LMTEMPLATE.COM (the template file associated with test description LMTEST1) can be rewritten using DTM\$TEST_NAME.

If you create a modified copy of LMTEST1 (the test description that previously used template file LMTEMPLATE.COM) and call the modified copy LMTEST2, you can easily generalize LMTEMPLATE.COM to run with both LMTEST1 and LMTEST2 by using DTM\$TEST_NAME in the template file. Here is the more general template file:

```
$ ! TEMPLATE for a simple test of the DSR left margin (.LM) command
$ !
$ ! Display the input file
$ !
$
$ TYPE 'dtm$test_name'.RNO
$
$ !
$ ! Run DSR on the input file
$ !
$
$ DSR 'dtm$test_name'.RNO
$
$ !
$ ! Show the output of this runoff test
$ !
$
$ TYPE 'dtm$test_name'.MEM
$
$ DELETE 'dtm$test_name'.MEM;1
```

This new template file can be used with any DSR tests whose test name is the same as that of the RNO input file used to exercise the DSR software, that is, this template file can be used by test description LEFTMARGIN2 with test input file LEFTMARGIN2.RNO, and by test description PAGE1 with test input file PAGE1.RNO, and so on.

6.3.3 DTM\$COLLECTION_NAME Global String Symbol

DEC/Test Manager defines the VAX/VMS global string symbol DTM\$COLLECTION_NAME before the collection prologue file executes. It is available for use in any prologue, epilogue, or template file in the collection. Many DEC/Test Manager users want a quick report of the status of a collection at the end of the run and to be informed when the collection is finished. You can do this by having the collection prologue file do the following:

- Invoke DEC/Test Manager.
- Enter the DTM SHOW COLLECTION/FULL command using DTM\$COLLECTION_NAME to specify the collection name.
- Invoke the VAX/VMS MAIL Utility to send you the output of this command.

For example:

```
$ ! This collection epilogue file mails a copy of the completed collection
$ ! run information to the project account.
$ !
$ DTM SHOW COLLECTION 'dtm$collection_name'/FULL -
$ /OUTPUT='dtm$collection_name'.REPORT
$ MAIL 'dtm$collection_name'.REPORT/sub="collection summary" DSRPROJECT
$ DELETE 'dtm$collection_name'.REPORT;
$ !
```

6.4 Grouping Tests

DEC/Test Manager provides a feature that lets you classify test descriptions by placing them into user-created categories called **groups**. Each group in the library is identified with a **group-name**, which is unique in the current library. This section presents the DEC/Test Manager commands used to create groups, to change the contents of groups, to delete groups, and to build a hierarchy of groups.

Groups are an optional and convenient feature of DEC/Test Manager. On the DEC/Test Manager commands that accept the test-group-expression parameter, you can specify a set of test descriptions by specifying their group-name. This saves you the trouble of specifying a long list of test-names. Using groups lets you organize your test descriptions according to common attributes. A number of tests that test one command, or a number of tests created during a particular period of time, can be grouped together.

For example, you could create a group called LEFTMARGINS that contains three tests of the LEFT MARGIN command:

```
LEFTMARGINS
  LMTEST1
  LMTEST2
  LMTEST3
```

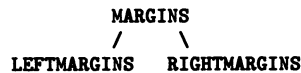
You could create a group called NEWTESTS that contains recently created tests:

```
NEWTESTS
  LMTEST1
  RMTEST1
  RMTEST2
```

Note that a test can belong to more than one group. In this case, the test LMTEST1 belongs to both groups.

You place test descriptions into groups by specifying their individual test-names, by specifying a wildcard form of a test-name, or by specifying a group-name (to include all the test descriptions in a group).

You construct a **group hierarchy** when you insert groups into other groups. For example, you can construct a set of groups that can be represented as follows:



In this example, groups LEFTMARGINS and RIGHTMARGINS are **subgroups** or **members** of the group MARGINS because both groups are now contained in MARGINS. Whenever you specify group MARGINS in a test-group-expression, you are specifying the contents of groups LEFTMARGINS and RIGHTMARGINS.

You can insert a test into a single group or into several groups. You can insert a group into a single group or into several groups. However, you cannot create a recursive group; that is, you cannot insert a group into itself or into a group to which it already belongs. For example, because RIGHTMARGINS is a subgroup of MARGINS, you cannot insert MARGINS into RIGHTMARGINS.

6.4.1 Creating Groups—CREATE GROUP

Before you can insert test descriptions (or groups) into a group, you must create the group-name in the DEC/Test Manager library with the CREATE GROUP command. The format is

```
DTM CREATE GROUP group-name [remark]
```

The group-name parameter specifies a unique name for the group. You can create only one group at a time.

The CREATE GROUP command takes the /[NO]LOG qualifier.

The following command creates the group MARGINS and includes a comment:

```
‡ DTM CREATE GROUP MARGINS "tests for margin setting commands"
```

6.4.2 Inserting Test Descriptions into Groups—INSERT TEST_DESCRIPTION

Use the INSERT TEST_DESCRIPTION command to place test descriptions in a group. A test description inserted into a group with this command remains in the group until you remove it with the REMOVE TEST_DESCRIPTION command.

The format is

```
DTM INSERT TEST_DESCRIPTION test-group-expression group-expression [remark]
```

The test-group-expression parameter specifies the tests to be included in the group. This parameter can include test-names and group-names, wildcard characters, wildcard characters in combination with partial or full test-names and group-names, or a list of these separated by commas. Identify the items in the test-group-expression by following each with the appropriate parameter qualifier, /GROUP or /TEST_DESCRIPTION. The default qualifier is /TEST_DESCRIPTION.

The INSERT TEST_DESCRIPTION command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following examples show three ways to use test-group-expressions to insert test descriptions into a group. If you have three tests with test-names LMTEST1, LMTEST2, and LMTEST3, use the following command to add the three tests to the group LEFTMARGINS. The first command lists the three tests by test-name. Because the parameter qualifier /TEST_DESCRIPTION is the default, you do not need to add it after each item.

```
⌘ DTM INSERT TEST_DESCRIPTION LMTEST1,LMTEST2,LMTEST3 LEFTMARGINS
```

The second command uses a wildcard character (*) in combination with a partial test-name:

```
⌘ DTM INSERT TEST_DESCRIPTION LMTEST* LEFTMARGINS
```

The third command uses another wildcard character (%) in combination with a more complete test-name:

```
⌘ DTM INSERT TEST_DESCRIPTION LMTEST% LEFTMARGINS
```

All three commands insert the tests (LMTEST1, LMTEST2, and LMTEST3) into the group LEFTMARGINS.

The following examples show two ways to add the tests to another group, SMITHTESTS, using test-group-expressions. If the tests are already contained in group LEFTMARGINS and if they are the only test descriptions in LEFTMARGINS, you can insert these tests into SMITHTESTS using a group-expression. Insert the test descriptions into SMITHTESTS using the group-expression LEFTMARGINS:

```
⌘ DTM INSERT TEST_DESCRIPTION LEFTMARGINS/GROUP SMITHTESTS
```

The parameter qualifier /GROUP identifies LEFTMARGINS to DEC/Test Manager as a group and indicates that the three tests in LEFTMARGINS are to be inserted into SMITHTESTS.

The following command adds the tests to both groups, LEFTMARGINS and SMITHTESTS, at the same time using the group-expression SMITHTESTS,LEFTMARGINS, where both groups are specified as a list:

```
⌘ DTM INSERT TEST_DESCRIPTION LMTEST* SMITHTESTS,LEFTMARGINS
```

When to Use the INSERT TEST_DESCRIPTION Command

You can use the INSERT TEST_DESCRIPTION command if, for example, you want to ensure that you use the test description LMTEST1 whenever you specify the group MARGINS. If LMTEST1 is already part of another group, LEFTMARGINS, you can use the INSERT GROUP command to insert the group LEFTMARGINS into the group MARGINS. However, this command does not place the test description LMTEST1 directly into the group MARGINS. If LMTEST1 is removed from the LEFTMARGINS group, or if the group LEFTMARGINS is removed from the group MARGINS, LMTEST1 is no longer specified when the group MARGINS is specified.

To place LMTEST1 directly into the group MARGINS, use the INSERT TEST_DESCRIPTION command. Then, whenever you specify the group MARGINS, you also specify LMTEST1, until you remove the test description from MARGINS with the REMOVE TEST_DESCRIPTION command.

6.4.3 Inserting Groups into Other Groups—INSERT GROUP

You can build hierarchies of groups (group trees) in the library by including one group within another using the INSERT GROUP command, if you do not create recursive groups. The inserted group becomes a **subgroup** of the other group. Remove a subgroup from another group using the REMOVE GROUP command.

Creating a group hierarchy by inserting groups into other groups builds generality into your group structure. Any changes you make to the subgroup directly affect the group and all other groups to which this subgroup belongs.

The format is

```
DTM INSERT GROUP group-expression1 group-expression2 [remark]
```

The group-expression1 parameter specifies the groups to be inserted. The group-expression2 parameter specifies the groups that are to contain the specified groups. The group-expression parameters can include group-names, wildcard characters, wildcard characters in combination with partial or full group-names, or a list of these separated by commas. The INSERT GROUP command takes all groups specified in group-expression1 and inserts them into all groups specified in group-expression2.

The INSERT GROUP command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following command inserts the group RIGHTMARGINS into the group MARGINS:

```
‡ DTM INSERT GROUP RIGHTMARGINS MARGINS
```

From now until you remove RIGHTMARGINS from MARGINS, whenever you specify MARGINS you specify all the test descriptions currently in its subgroup RIGHTMARGINS, as well as any other test descriptions or groups contained in MARGINS. For example, if you specify MARGINS when you create a collection, all test descriptions in the subgroup RIGHTMARGINS and all other test descriptions in MARGINS are included in the collection.

It is important to note that the contents of a group change as the contents of its subgroups change. For example, if RIGHTMARGINS contains a test description with test-name RMTEST1, RMTEST1 is specified whenever MARGINS is specified. However, if RMTEST1 is removed from RIGHTMARGINS, it is no longer specified by the group MARGINS.

The following example shows a group-expression that contains wildcard characters to add the groups LEFTMARGINS and RIGHTMARGINS to the group MARGINS. As a precaution, you should first use the SHOW GROUP command to verify that these are the only groups identified by the wildcard expression.

```
‡ DTM SHOW GROUP *TMARGINS
```

```
Groups in DEC/Test Manager library DISK$USER01:[PROJECT.DTMLIB]  
LEFTMARGINS  
RIGHTMARGINS
```

```
‡ DTM INSERT GROUP *TMARGINS MARGINS
```

When to Use the INSERT GROUP Command

You can use the INSERT GROUP command if, for example, you want to ensure that the group MARGINS always contains all the tests in the LEFTMARGINS and RIGHTMARGINS groups. You can do this by inserting all the test descriptions in LEFTMARGINS and RIGHTMARGINS into MARGINS with the INSERT TEST_DESCRIPTION command. You should use the SHOW GROUP command to confirm that the contents of MARGINS matches the contents of LEFTMARGINS and RIGHTMARGINS.

A more efficient way to ensure that the tests in MARGINS match the tests in LEFTMARGINS and RIGHTMARGINS is to place the LEFTMARGINS and RIGHTMARGINS groups into the MARGINS group with the INSERT GROUP command. Then, whenever you specify MARGINS, DEC/Test Manager checks RIGHTMARGINS and LEFTMARGINS and includes their current contents as part of the MARGINS group.

6.4.4 Examining the Contents of Groups—SHOW GROUP

Use the SHOW GROUP command to view a library's current group structure. The format is

```
DTM SHOW GROUP group-expression
```

The group-expression parameter can include group-names, wildcard characters, wildcard characters in combination with partial or full group-names, or a list of these separated by commas.

The SHOW GROUP command takes the following qualifiers:

```
/BRIEF  
/[NO]CONTENTS  
/FULL  
/INTERMEDIATE  
/MEMBER  
/OUTPUT
```

The qualifiers /BRIEF, /INTERMEDIATE, /FULL, and /CONTENTS control the amount of information displayed for each group. The default, /INTERMEDIATE, displays the group-name and any associated remark. The /CONTENTS=n qualifier specifies the number of levels of contents shown for a group. The default is 1. Specify the /CONTENTS=ALL qualifier or the /FULL qualifier to display the full contents of a group. The /MEMBER qualifier displays the group-names of all groups to which the specified group belongs.

The following example shows a SHOW GROUP command for the group MARGIN_TESTS, which contains two subgroups (LEFTMARGINS and RIGHTMARGINS) that were added to it with the INSERT GROUP command, and a test description (MARGINTEST1) that was added to it with the INSERT TEST_DESCRIPTION command. The /CONTENTS=ALL qualifier also displays the current contents of LEFTMARGINS and RIGHTMARGINS.

```

$ DTM SHOW GROUP MARGIN_TESTS /CONTENTS=ALL
Groups in DEC/Test Manager Library DRA1: [PROJECT.DTMLIB]
MARGIN_TESTS      "tests for line margin setting commands"
  LEFTMARGINS/Group
    LMTEST1
    LMTEST2
    LMTEST3
  RIGHTMARGINS/Group
    RMTEST1
    RMTEST2
    RMTEST3
  MARGINTEST1

```

The SHOW TEST_DESCRIPTION/FULL command lists the groups to which a specified test description belongs.

6.4.5 Removing the Contents of Groups—REMOVE GROUP and REMOVE TEST_DESCRIPTION

Use the REMOVE TEST_DESCRIPTION and REMOVE GROUP commands to remove test descriptions and subgroups from groups. The REMOVE commands dissolve group associations, but they do not delete tests or groups from the DEC/Test Manager library. Because a group must be empty before it can be deleted and a test description must not belong to any group before it can be deleted, you may need to use these REMOVE commands before deleting test descriptions or groups.

The REMOVE TEST_DESCRIPTION command removes a test description from one or more groups. It reverses the action of the INSERT TEST_DESCRIPTION command. The format is

```
DTM REMOVE TEST_DESCRIPTION test-group-expression group-expression [remark]
```

The test-group-expression parameter specifies the tests and groups to be removed from the groups. This parameter can include test-names and group-names, wildcard characters, wildcard characters in combination with partial or full test-names and group-names, or a list of these separated by commas. Identify the items in the test-group-expression by following each with the appropriate parameter qualifier, /GROUP or /TEST_DESCRIPTION. The default is /TEST_DESCRIPTION. The group-expression parameter specifies the groups from which the tests and groups are to be removed. This parameter can include group-names, wildcard characters, wildcard characters in combination with partial or full group-names, or a list of these separated by commas.

The REMOVE TEST_DESCRIPTION command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following command removes LMTEST1 from LEFTMARGINS without prompting you for confirmation:

```
⚡ DTM REMOVE TEST_DESCRIPTION LMTEST1 LEFTMARGINS /NOCONFIRM
```

The following command removes all test descriptions from LEFTMARGINS without prompting you for confirmation:

```
⚡ DTM REMOVE TEST_DESCRIPTION * LEFTMARGINS /NOCONFIRM
```

The REMOVE GROUP command removes one or more subgroups from one or more specified groups. It reverses the action of the INSERT GROUP command. The format is

```
DTM REMOVE GROUP group-expression1 group-expression2 [remark]
```

The group-expression1 parameter specifies the subgroups to be removed. The group-expression2 parameter specifies the groups from which the subgroups are to be removed. The group-expression parameters can include group-names, wildcard characters, wildcard characters in combination with partial or full group-names, or a list of these separated by commas.

The REMOVE GROUP command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following command removes all subgroups from MARGINS without prompting you for confirmation:

```
⚡ DTM REMOVE GROUP * MARGINS /NOCONFIRM
```

6.4.6 Deleting Groups from the Library—DELETE GROUP

The DELETE GROUP command deletes a group from the DEC/Test Manager library. Before deleting a group, you must remove its contents and its associations with other groups with the REMOVE GROUP command or the REMOVE TEST_DESCRIPTION command. If a group is empty and is not a subgroup of other groups, you can delete it from the DEC/Test Manager library with the DELETE GROUP command.

The format is

```
DTM DELETE GROUP group-expression [remark]
```

The group-expression parameter specifies the groups to be deleted. The group-expression parameter can include group-names, wildcard characters, wildcard characters in combination with partial or full group-names, or a list of these separated by commas.

The DELETE GROUP command takes the following qualifiers:

```
/[NO]CONFIRM  
/[NO]LOG
```

The following example deletes the group LEFTMARGINS from the DEC/Test Manager library:

1. Remove all test descriptions from LEFTMARGINS.

```
DTM REMOVE TEST_DESCRIPTION * LEFTMARGINS /NOCONFIRM
```

2. Remove all groups from LEFTMARGINS.

```
‡DTM REMOVE GROUP * LEFTMARGINS /NOCONFIRM
```

3. Remove LEFTMARGINS from all groups.

```
‡DTM REMOVE GROUP LEFTMARGINS * /NOCONFIRM
```

4. Delete LEFTMARGINS from the DEC/Test Manager library.

```
‡DTM DELETE GROUP LEFTMARGINS /NOCONFIRM
```

Note that it is unnecessary to follow all these steps for every case in which you want to delete a group. You need only use the commands necessary to dissolve the group you have built. The remainder of this section is an example that shows how you might create a group hierarchy for the sample DSR library. The groups are arranged so that the larger groups

(groups that contain subgroups) have a broader, more general meaning, while the subgroups contain test descriptions that test specific commands for the sample software project DSR.

The following groups are created:

LEFTMARGINS	Contains all test descriptions for tests that exercise the leftmargin command.
RIGHTMARGINS	Contains all test descriptions for tests that exercise the rightmargin command.
ALIGNMENT	Contains all test descriptions for tests that exercise commands performing special margin alignment.
MARGINS	Contains all groups containing test descriptions for general margin tests.
PAGING	Contains all groups containing test descriptions for tests that exercise page formatting commands (that is, all previously mentioned groups).

Notice that a test description can belong to more than one group. The following steps show a way to construct this hierarchy. This example assumes that you have already created the test descriptions for the nine tests with test-names LMTEST1 through LMTEST3, RMTEST1 through RMTEST3, and PGTEST1 through PGTEST3 in the current DEC/Test Manager library.

1. The following commands create the groups:

```
⌘ DTM CREATE GROUP RIGHTMARGINS "tests exercising the .RM command" /NOLOG
⌘ DTM CREATE GROUP LEFTMARGINS "all tests involving the .LM command" /NOLOG
⌘ DTM CREATE GROUP ALIGNMENT "various tests of alignment command" /NOLOG
⌘ DTM CREATE GROUP PAGING "tests of all page boundary commands" /NOLOG
⌘ DTM CREATE GROUP MARGINS "tests for margin setting commands" /NOLOG
```

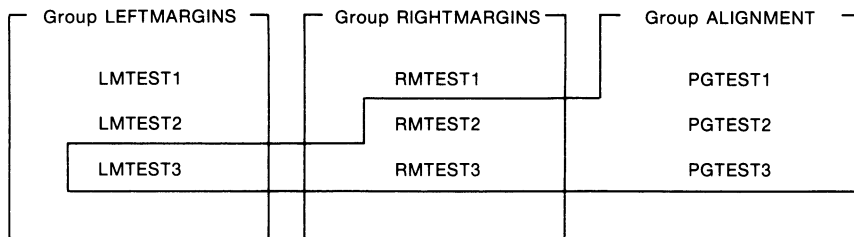
2. The following commands insert the test descriptions into the specified groups. Using wildcards, these commands insert all tests beginning with LM into the group LEFTMARGINS, all tests beginning with RM into the group RIGHTMARGINS, and tests pertaining to alignment into the group ALIGNMENT:

```
⌘ DTM INSERT TEST_DESCRIPTION LM* LEFTMARGINS /NOLOG
⌘ DTM INSERT TEST_DESCRIPTION RM* RIGHTMARGINS /NOLOG
⌘ DTM INSERT TEST_DESCRIPTION PG*,RMTEST2,RMTEST3,LMTEST3 ALIGNMENT /NOLOG
```

Figure 6-1 shows the group hierarchy constructed thus far.

Figure 6-1: Inserting Test Descriptions into Groups

Test descriptions inserted into groups.



ZK-2080-84

3. The following commands continue to build the group hierarchy by inserting various subgroups into other groups:

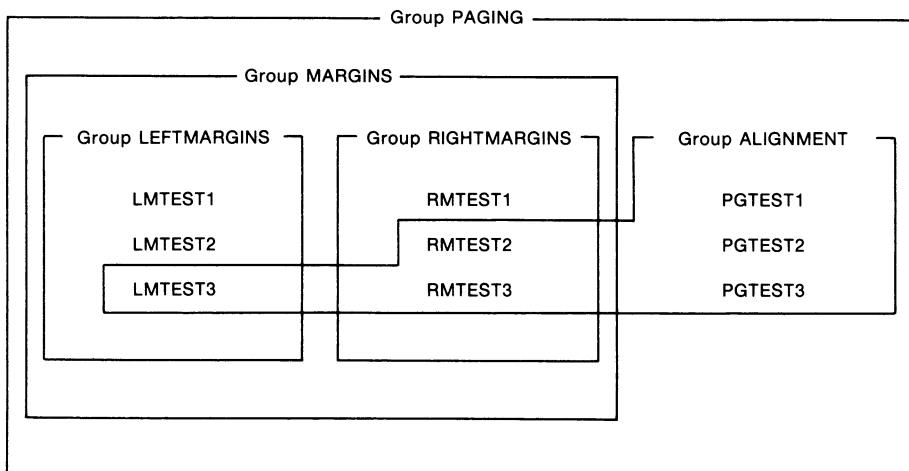
```
‡ DTM INSERT GROUP LEFTMARGINS,RIGHTMARGINS MARGINS /NOLOG  
‡ DTM INSERT GROUP MARGINS,ALIGNMENT PAGING /NOLOG
```

LEFTMARGINS and RIGHTMARGINS are now contained in the MARGINS group, which, along with ALIGNMENT, is now contained in the PAGING group.

Figure 6-2 shows the group hierarchy built with these commands.

Figure 6-2: Inserting Groups into Groups

Groups inserted into groups.



ZK-2079-84

Note that because the group LEFTMARGINS, for example, is a subgroup of both MARGINS and PAGING, when the contents of LEFTMARGINS change, the contents of both MARGINS and PAGING also change.

4. Use the SHOW GROUP command to examine the structure of a group. The following command shows the full contents of the group PAGING:

```
§ DTM SHOW GROUP/CONTENTS=ALL PAGING
```

Groups in DEC/Test Manager Library DRA1: [PROJECT.DTMLIB]

```
PAGING      "tests of all page boundary commands"
            ALIGNMENT/Group
                LMTEST3
                PGTEST1
                PGTEST2
                PGTEST3
                RMTEST2
                RMTEST3
            MARGINS/Group
                LEFTMARGINS/Group
                    LMTEST1
                    LMTEST2
                    LMTEST3
                RIGHTMARGINS/Group
                    RMTEST1
                    RMTEST2
                    RMTEST3
```

6.5 Using the Keypads

DEC/Test Manager supplies you with five keypads:

- A keypad that is available from the DEC/Test Manager subsystem—the DEC/Test Manager keypad.
- A keypad with default keypad definitions that is available from the Review subsystem—the Review subsystem keypad shown in Figure 6-3.
- A keypad to assist you in reviewing the result and benchmark files for interactive tests—the Review SHOW/RESULTS and SHOW/BENCHMARK keypad shown in Figure 6-4. The use of this keypad is described in Chapter 5.
- A keypad to assist you in reviewing the differences file for interactive tests—the Review SHOW/DIFFERENCES keypad shown in Figure 6-5. The use of this keypad is described in Chapter 5.
- A keypad to assist you in displaying the benchmark file for interactive tests—the DISPLAY/BENCHMARK keypad. This keypad is identical to the Review SHOW/RESULTS and SHOW/BENCHMARK keypad shown in Figure 6-4. The use of this keypad is described in Chapter 3.

The DEC/Test Manager keypad is available when you are using DEC/Test Manager as a subsystem. The GOLD key (PF1), the HELP key (PF2), and the ENTER (RETURN) key are defined on this keypad.

Most of the keypad keys on the DEC/Test Manager and Review subsystem keypads can be defined to execute two DEC/Test Manager commands. To execute the first command associated with the keypad key, press the appropriate keypad key. To execute the second command, press PF1 (the GOLD key) and then press the keypad key.

For the DEC/Test Manager and Review subsystem keypads, you can redefine the keys that have default definitions and you can define the previously undefined keys. You cannot change the key definitions of the Review SHOW and DISPLAY/BENCHMARK command keypads.

Use the DEFINE/KEY command to define keys on the DEC/Test Manager keypad and on the Review subsystem keypad. The formats are

```
DTM> DEFINE/KEY key-name "command-string"
```

```
DTM_REVIEW> DEFINE/KEY key-name "command-string"
```

The key-name parameter specifies the key to define. The command-string parameter is the command to associate with the command. If the command contains spaces, enclose it in double quotation marks ("").

The DEFINE/KEY command takes the following qualifiers:

```
/[NO]ECHO  
/[NO]IF_STATE  
/[NO]LOCK_STATE  
/[NO]SET_STATE  
/[NO]TERMINATE
```

The /[NO]ECHO qualifier specifies whether the command is to be echoed when the key is pressed. The /[NO]TERMINATE qualifier specifies whether you must press RETURN after pressing the keypad key to execute the command. You cannot define a keypad key as both /NOECHO and /NOTERMINATE. You cannot define a keypad key as both /TERMINATE and /SET_STATE.

The /[NO]IF_STATE qualifier specifies the state (if there is one) that is to be associated with the keypad definition. The /[NO]LOCK_STATE qualifier specifies whether the state specified with /SET_STATE should be retained until another /SET_STATE is established. You cannot define a keypad key as both /TERMINATE and /SET_STATE. The DTM and GOLD_DTM states are associated with the DTM keypad. The REVIEW and GOLD_REVIEW states are associated with the Review subsystem keypad. When you associate the DTM or REVIEW state with a key on the respective keypad, you press only the keypad key to execute the command associated with the key. When you associate the GOLD_DTM

or GOLD_REVIEW state with a key, you must press the GOLD key (PF1) before pressing the keypad key to execute the command associated with the key.

The following commands define the previously undefined KP3 key on the DEC/Test Manager keypad to perform the SET LIBRARY DBA\$:[project.mydsrlib] command to select another DEC/Test Manager library:

```
DTM> DEFINE/KEY KP3 "SET LIBRARY DRA$:[SMITH.MYLIB]" /TERMINATE
DTM>
```

If you then press KP3, you see the following:

```
DTM> SET LIBRARY DRA$:[SMITH.MYLIB]
%DTM-S-LIBIS, DEC/Test Manager library is [SMITH.MYLIB]
DTM>
```

The following commands show how to use the Review subsystem DEFINE/KEY command to redefine a key on the Review keypad. They define the gold function associated with the KP5 key to spawn MAIL as a subprocess.

```
DTM_REVIEW> DEFINE/KEY KP5 /IF_STATE=GOLD_REVIEW "SPAWN MAIL"
DTM_REVIEW>
```

Pressing PF1 followed by KP5 causes DEC/Test Manager to create a subprocess and execute the DCL MAIL command:

```
DTM_REVIEW> SPAWN MAIL
MAIL>
```

Type the EXIT command to leave the MAIL Utility and return to your Review session.

When you create key definitions with the DEFINE/KEY command, these definitions are in effect only for the current DEC/Test Manager session. The next time you invoke DEC/Test Manager, only the default key definitions will be in effect. To save your key definitions and to use them in every DEC/Test Manager session you initiate, include your key definitions in a DEC/Test Manager initialization file. This file is executed whenever you invoke DEC/Test Manager as a subsystem. For more information on initialization files, see Section 6.7.1.

If you have key definitions that you want to save but do not necessarily want to use every time you invoke DEC/Test Manager, store them in a command procedure. For more information on DEC/Test Manager command procedures, see Section 6.7.

Figure 6-3: Default Review Subsystem Keypad

REVIEW
GOLD_REVIEW

PF1 GOLD	PF2 HELP KEYPAD (HELP)	PF3	PF4
7 SHO/DIF PRI/DIF	8 SHO/RES PRI/RES	9 SHO/BEN PRI/BEN	— SHO/SUM
4 PRI/NOW	5	6 UPDATE	, SPAWN ATT/PAR
1 NEXT/UN BACK/UN	2 NEXT/NE BACK/NE	3 NEXT/UP BACK/UP	ENTER
0 NEXT BACK		• FIRST LAST	

The RETURN key and the ENTER key move you to the next result description.

ZK-4749-85

Figure 6-4: Review SHOW/RESULT, Review SHOW/BENCHMARK, and DISPLAY/BENCHMARK Keypad

PF1	PF2 HELP KEYPAD	PF3	PF4
7 SCREEN NUMBER DISPLAY	8	9	—
4	5 FIRST SCREEN	6	,
1 PREV SCREEN	2	3	ENTER
0 NEXT SCREEN		.	

ZK-4750-85

Figure 6-5: Review SHOW/DIFFERENCES Keypad

PF1	PF2 HELP KEYPAD	PF3	PF4
7 SCREEN NUMBER DISPLAY	8 SPLT-SCR MODE	9 FULL-SCR MODE	—
4 HIGHLIGHT DIFFERENCES	5 FIRST SCREEN	6	,
1 PREV SCREEN	2	3 SHIFT SPLT-SCR MODE	ENTER
0 NEXT SCREEN		.	

ZK-4751-85

6.6 Correcting Errors in the DEC/Test Manager Library—VERIFY and VERIFY/RECOVER

The VERIFY command performs a series of checks on the DEC/Test Manager library to make sure that it has a valid structure and that files in the library are valid. It also consolidates memory. If the library is valid, the command completes successfully. If not, DEC/Test Manager instructs you to use the VERIFY/RECOVER command to correct errors in the library and to return it to a usable state. The format is

DTM VERIFY

The VERIFY command takes the following qualifiers:

```
/[NO]LOG  
/RECOVER
```

The VERIFY/RECOVER command attempts to restore an unusable DEC/Test Manager library to a useful state. If the VERIFY/RECOVER command fails to return your library to a usable state, restore it from a backup tape.

You should enter the VERIFY/RECOVER command if instructed to do so in response to a DEC/Test Manager error message. This command does not harm a properly created DEC/Test Manager library. If the VERIFY/RECOVER command discovers subdirectories that are not associated with a current collection while restoring the library, you are prompted before the directories are deleted. You should not create subdirectories in the DEC/Test Manager library.

If a system failure occurs while a DEC/Test Manager command that alters the library is executing, the library is placed in an unusable state. The VERIFY/RECOVER command cleans up the library and deletes partially complete transactions.

If a collection run is terminated other than with the DTM STOP command or by pressing CTRL/C, that collection will be locked and you will not be able to review it. In this case, VERIFY/RECOVER cleans up the library and marks the tests that did not run. After entering the VERIFY/RECOVER command, compare and then review the collection. While reviewing the collection, you can create a group containing the tests that did not run. After leaving the Review subsystem, you can execute that group of tests. It is sometimes necessary to enter the VERIFY/RECOVER command twice to restore a collection after aborting a collection run.

6.7 Using Command Files

A DEC/Test Manager command file or command procedure is a file containing one or more DEC/Test Manager commands. You create DEC/Test Manager command files with a text editor and execute them by entering the @file-spec command, where file-spec is the file specification for the command file. If you include only a file name, DEC/Test Manager assumes the COM file type. You can invoke a command file from the DEC/Test Manager subsystem level, from the Review subsystem level, or from within another command file. When you invoke a command file, its

commands execute in sequence. If the command file causes an error or warning to occur, execution of the command file stops and no subsequent commands are executed.

If a second @file-spec command is encountered within the command file, the second command file is invoked and its commands are executed sequentially. After all the commands in the second command file are executed, control returns to the first command file. DEC/Test Manager places no restrictions on the number of command files you can nest. DEC/Test Manager also does not check for recursive command files. When the last command in the initial command file is executed, control is left at the current level. An EXIT command causes DEC/Test Manager to terminate the subsystem that is running. It does not necessarily terminate execution of the command file. If an EXIT command terminates the DEC/Test Manager subsystem, execution of the command file is also terminated.

When you create a command file, each command you enter must be entered on a single line. You cannot use a hyphen to continue commands onto a second line. Enter comments into your command file by placing an exclamation mark (!) in column 1.

The logical name DTM\$INIT is a DEC/Test Manager-provided logical name that you define to identify a command file that you want DEC/Test Manager to execute each time you invoke DEC/Test Manager. For more information, see Section 6.7.1.

When you invoke DEC/Test Manager in subsystem mode from a DCL command procedure, be sure to supply all required command parameters. If you omit a required parameter for which you would be prompted if you entered the command interactively, DCL reads the next line in the command file as the missing parameter rather than as a separate command. The second command is lost.

If you omit a required parameter on a DEC/Test Manager command in a DEC/Test Manager command procedure, DEC/Test Manager prompts you for the missing parameter.

6.7.1 Using a DEC/Test Manager Initialization Command File

DEC/Test Manager provides the VAX/VMS logical name DTM\$INIT. Use the following command to define it to identify your DEC/Test Manager initialization command file:

```
⌘ DEFINE DTM$INIT file-spec
```

File-spec is the file specification for your initialization command file, a command file that is executed whenever you invoke DEC/Test Manager without the /NOINIT qualifier. A typical initialization command file would contain the commands you enter every time you invoke DEC/Test Manager. For example, it could contain the command to select a DEC/Test Manager library and the commands to define keys on the DEC/Test Manager keypad.

The following example establishes the DEC/Test Manager library as [PROJECT.DSR] and defines several additional keypad keys on the DEC/Test Manager keypad.

```
!Initialization file to set library and define keys
!
!Establish the library
!
SET LIBRARY DRA$:[PROJECT.DSR]
!
!Define keypad keys
!
DEFINE/KEY KP3/IF_STATE=DTM           "SHOW COLLECTION */FULL" /TERMINATE
DEFINE/KEY PF4/IF_STATE=GOLD_DTM      "SET LIBRARY DRA$:[SMITH.MYLIB]" /TERMINATE
DEFINE/KEY KP1/IF_STATE=REVIEW        "NEXT/SUCCESSFUL" /TERMINATE
DEFINE/KEY KP1/IF_STATE=GOLD_REVIEW   "BACK/SUCCESSFUL" /TERMINATE
```

6.8 Spawning or Attaching to Another Process—SPAWN and ATTACH

Both the DEC/Test Manager subsystem and the Review subsystem have SPAWN and ATTACH commands. These commands allow you to create one or more subprocesses of your parent process, and to move between these processes.

The SPAWN command allows you to create, or spawn, a subprocess and to attach your terminal to it. You can create a subprocess to issue DCL commands, to read an electronic mail message, or to create another DEC/Test Manager session. The ATTACH command allows you to switch between your subprocesses.

If you specify a DCL command as a parameter to the SPAWN command, the DCL command is executed and control is returned immediately to the DEC/Test Manager session. If you do not include a DCL command, the DCL prompt displays, and you can then issue DCL commands. As each command terminates, the DCL prompt is redisplayed. You can return to your parent process by logging out of the subprocess or by issuing the ATTACH command.

6.9 Using Filters

DEC/Test Manager provides six filters to mask data in result files that varies from one test run to the next. These filters are run only when the test is included in a collection and executed. Filters are not run when a test is recorded or played. The filters execute after the test epilogue file. This sequence is shown in Figure 4-1. The DEC/Test Manager filters are described in Table 6-1.

Table 6-1: Filters Provided by DEC/Test Manager

Filter	Description
ALL	Specifies all six filters are to be used
DATE	Replaces date stamps: of the form 1-OCT-86 with dd-mmm-yyyy of the form 11-OCT-86 with dd-mmm-yyyy of the form 9/23/86 with mm/dd/yy of the form September 10, 1986 with month day, year
DIRECTORIES	Replaces directory specifications with DISK:[DIRECTORY]
FILE_NAMES	Replaces file names with FILENAME.EXT
TIME	Replaces time stamps with hh:mm:ss.xxxx
TRACE_BACK	Replaces the 8-bit memory address with xxxxxxxx
VERSION	Replaces file versions with VERSION

To use a filter on a test's result description, you must associate the filter with the test. Do this by including the filter in the test description filter's field. For a new test description, use the CREATE TEST_DESCRIPTION/FILTER command and list the filters you want associated with the test. For an existing test description, first use the MODIFY TEST_DESCRIPTION/FILTER command and list the filters you want associated with the test. Then use the RECREATE command to recreate any collections containing the test.

The `MODIFY TEST_DESCRIPTION/NOFILTER` command dissociates the specified filters from the test description. You can remove all or some of the filters from the test description. The `COPY TEST_DESCRIPTION/NOFILTERS` command dissociates all filters from the new test description you are creating. A test description created with the `COPY TEST_DESCRIPTION` command either has all the filters associated with it that were associated with the existing test description, or it has no filters associated with it.

The `SHOW TEST DESCRIPTION/FULL` and `SHOW TEST DESCRIPTION/FILTER` commands list the filters associated with a specific test description.

NOTE

Exercise caution when using filters. The original unfiltered result file is deleted after the filtering occurs, leaving only the filtered file. Using some filters on interactive tests that contain escape sequences can delete information that is essential to the test.

6.10 History Logging

Whenever you issue a DEC/Test Manager command that alters the library, DEC/Test Manager logs that command and its associated remark in a chronological history file. This section describes how to look at the information contained in the history file and how to enter a comment into the history file.

6.10.1 Displaying the History File—SHOW HISTORY

The DEC/Test Manager `SHOW HISTORY` command displays a chronological list of library transactions. The commands that are logged are listed in the description of the `SHOW HISTORY` command in the Command Dictionary. The format is

```
DTM SHOW HISTORY object-expression
```

Object-expression is the DEC/Test Manager entity whose history you want to see. An object-expression can be one or more group-names, test-names, collection-names, or wildcard forms of these.

The SHOW HISTORY command takes the following qualifiers:

```
/BEFORE  
/OUTPUT  
/SINCE  
/[NO]TRANSACTION  
/USER
```

The /BEFORE and /SINCE qualifiers display history information recorded before or after a specified time. The /TRANSACTIONS qualifier displays history information about the specified commands. The /USER qualifier displays history information about all commands entered by a specified user.

6.10.2 Entering Remarks on DEC/Test Manager Commands

All DEC/Test Manager commands that alter the library in any way accept an optional remark. If you enter a remark, it is logged in the history file. Remarks are useful for tracking changes made to the DEC/Test Manager library.

If you do not enter a remark, DEC/Test Manager prompts you for one.

Remark:

If you press RETURN instead of entering a remark, DEC/Test Manager logs a null remark ("") with the command.

When you enter a remark on the same line as the command, enclose the remark within double quotation marks (""). When you enter the remark in response to the Remark: prompt, do not enter the quotation marks.

You can also use the DTM REMARK command to enter remarks directly into the history file. You may want to use this command to document an unusual occurrence in the library, or to indicate why you deleted information from the history file.

6.11 Displaying Information About the Current DEC/Test Manager Library—SHOW ALL

The SHOW ALL command displays a summary description about the current DEC/Test Manager library. The format is

```
DTM SHOW ALL
```

The SHOW ALL command takes the /OUTPUT qualifier.

The SHOW ALL command displays the following information:

- The directory specification for the library
- The default directory specifications for the current benchmark and template directories
- The file specifications for the default collection prologue and epilogue files
- The number of collections, test descriptions, groups, and variables in the library

If one of these entities is not specified for the library, the message NONE SPECIFIED is displayed.

For example, if your current DEC/Test Manager library is DRA1:[PROJECT.DTMLIB], entering the SHOW ALL command displays the following output:

```
‡ DTM SHOW ALL
```

```
Description of DEC/Test Manager Library DRA1:[PROJECT.DTMLIB]
```

```
Default template directory: DRA1:[PROJECT.TEMPLATES] ""
Default benchmark directory: DRA1:[PROJECT.BENCHMARKS] ""
Default collection prologue: None Specified
Default collection epilogue: None Specified
Number of collections:      20
Number of test descriptions: 152
Number of groups:          18
Number of variables:       9
```


Using DEC/Test Manager with the VAX Performance and Coverage Analyzer

This chapter describes how to use DEC/Test Manager with the VAX Performance and Coverage Analyzer. It assumes that you are already familiar with both tools. It is not a substitute for the VAX Performance and Coverage Analyzer documentation.

The VAX Performance and Coverage Analyzer is a software development tool that gathers performance and coverage data for a program and processes and formats that data. The VAX Performance and Coverage Analyzer can be used with DEC/Test Manager to provide information on how well the tests within DEC/Test Manager covered the program being tested, as well as information on the performance of that program under various tests.

Before you begin using these two tools together, you should do the following:

- Verify that both DEC/Test Manager and the VAX Performance and Coverage Analyzer are installed on your system.
- Read the *Guide to VAX DEC/Test Manager* and the *VAX Performance and Coverage Analyzer User's Reference Manual*.
- Become familiar with the commands and functions of both DEC/Test Manager and the VAX Performance and Coverage Analyzer.
- Create a DEC/Test Manager test system.

7.1 The Benefits of Using These Two Tools Together

Using DEC/Test Manager with the VAX Performance and Coverage Analyzer enables you to evaluate how well your test system covers your program code and how well your program performs under your test system.

Gathering coverage data on your test system helps you ensure that your tests exercise all intended code paths. During the development cycle, checking your test system coverage helps you find untested code in your program. During the maintenance cycle, checking your test system coverage helps you verify that your tests exercise all new code written to correct problems in your program.

Gathering performance data on your program helps you evaluate how well your program performs under different conditions of user input. You can check your program's performance either during individual test executions or averaged over all the executions in your test system.

7.2 Using These Two Tools Together

This section describes the steps to follow when using DEC/Test Manager with the VAX Performance and Coverage Analyzer:

1. Link the program to be tested with the Collector of the VAX Performance and Coverage Analyzer.
2. Create a Collector initialization file containing the data collection commands.
3. Create a DEC/Test Manager collection prologue file pointing to the Collector initialization file.
4. Create a DEC/Test Manager collection of tests. Use the `/PROLOGUE` qualifier to specify the collection prologue file created in step 3.
5. Execute the DEC/Test Manager collection. DEC/Test Manager then invokes the Collector in batch and the Collector then gathers data from your program as DEC/Test Manager executes the tests.
6. Enter the DEC/Test Manager Review subsystem to review your test results and to invoke the Analyzer.

Each of these steps is described in greater detail in the following sections.

7.2.1 Linking Your Program with the Collector

Whenever you want to gather data on a program using the Collector, you must compile, link, and run the program as described in the *VAX Performance and Coverage Analyzer User's Reference Manual*. The linking procedure does not change when you use the Collector with DEC/Test Manager. The only difference is that instead of immediately running the Collector yourself, DEC/Test Manager invokes the Collector in batch and gathers data while running software tests over your program.

7.2.2 Creating a Collector Initialization File

When you want DEC/Test Manager to invoke the Collector in batch, you must provide a Collector initialization file. The VAX Performance and Coverage Analyzer must know the location of this file.

The Collector initialization file must contain all the commands you want to be issued to the Collector. It might contain only one command: the GO command. When this is the case, only PC (program counter) samples are gathered.

If you want other data gathered, specify the kinds of data you want gathered by including the proper Collector SET commands. Note that the last command entered into the Collector initialization file must be the GO command because this command signals the beginning of the data-gathering process.

You should not include certain Collector commands in the Collector initialization file when using DEC/Test Manager with the VAX Performance and Coverage Analyzer. These commands override the logical names that DEC/Test Manager specifies for the names of the collection run and the Collector data file. If you use either the SET RUN_NAME command to designate a collection run name or the SET DATAFILE command to designate a Collector data file name, the new values override the values DEC/Test Manager assigns.

You may want to override these values if, for example, you want to use a Collector data file with a name other than the name assigned by default. However, if you do override one of these values, you cannot invoke the Analyzer from the DEC/Test Manager Review subsystem. You must instead invoke the Analyzer directly, using the VAX Performance and Coverage Analyzer PCA command at DCL level.

Note that the VAX Performance and Coverage Analyzer typically uses the term *collection* to mean that the Collector is collecting or gathering data, and the term *collection run* to mean a single program execution when the Collector gathers data. DEC/Test Manager uses the term *collection* to refer to an assembly of tests to be executed.

7.2.3 Creating a DEC/Test Manager Collection Prologue File

The DEC/Test Manager collection prologue file is a user-written DCL command procedure that DEC/Test Manager runs before running a collection of software tests. Generally, the collection prologue file is used to set up or augment the environment in which all the test files run. Because DEC/Test Manager invokes the Collector in batch, you must include a line in your collection prologue file informing the VAX Performance and Coverage Analyzer of the name and location of the Collector initialization file.

Specify the Collector initialization file by defining the VAX Performance and Coverage Analyzer logical name PCAC\$INIT to be equivalent to the Collector initialization file specification in the DEC/Test Manager collection prologue file. Following is an example of such a collection prologue file in its simplest form:

```
#!      Collection prologue file for running the Collector in batch mode
#!
#!      define the Collector initialization file
#!
$ DEFINE PCAC$INIT DRA1:[PROJECT]MYINITFILE.PCAC
#!
#!      End of collection prologue file
$
```

If you typically use a DCL logical name for the executable image of the program you want to test, you can redefine that logical name in the collection prologue file to point to the image that has been linked with the Collector's shareable image.

For example, if you specify RUNOFF\$EXE to be equivalent to the executable image of your program, you can redefine RUNOFF\$EXE to be the Collector-linked version of your executable image in the collection prologue file. A collection prologue file making this assignment, as well as specifying the Collector initialization file, looks like the following example.

```

$!      Collection prologue file for running the Collector in batch mode
$!
$!      define the Collector initialization file
$!
$ DEFINE PCAC$INIT  DRA1:[PROJECT]MYINITFILE.PCAC
$!
$!      define the Collector-linked exe
$!
$ DEFINE RUNOFF$EXE DRA1:[PROJECT]PCA_RUNOFF.EXE
$!
$!      End of collection prologue file
$
$

```

By using this collection prologue file to specify all the information that the VAX Performance and Coverage Analyzer needs, you place the use of this tool on a single switch. Consequently, when you create a collection using CREATE COLLECTION/PROLOGUE to include the collection prologue file described previously, DEC/Test Manager invokes the VAX Performance and Coverage Analyzer. When you do not use the /PROLOGUE qualifier to include this collection prologue file, DEC/Test Manager does not invoke the VAX Performance and Coverage Analyzer and the Analyzer is not available from the Review subsystem.

7.2.4 Creating a DEC/Test Manager Collection

You must create a DEC/Test Manager collection to run DEC/Test Manager and one or more of your tests.

As mentioned in Section 7.2.3, you can use the CREATE COLLECTION/PROLOGUE command as a switch for whether or not you want to collect performance data while your tests are running. To use the VAX Performance and Coverage Analyzer, specify the prologue file that sets up and invokes the Collector. For example:

```

DTM> CREATE COLLECTION LM_COLLECT_1 LMTEST_1, LMTEST_2 -
_DTM> /PROLOGUE=DRA1:[PUBLIC]PCA_PROLOGUE
_REMARK: Invoking the PCA collector

```

This specifies that you want to gather data with the Collector.

```

DTM> CREATE COLLECTION RM_COLLECT_1 RMTEST*
_REMARK: Running tests without the collector

```

This specifies that you do not want to use the Collector with this collection.

7.2.5 Executing a DEC/Test Manager Collection

After you create a collection with a collection prologue file that sets up the VAX Performance and Coverage Analyzer environment, you are ready to execute the collection. DEC/Test Manager always invokes the VAX Performance and Coverage Analyzer in batch, regardless of whether you execute the collection in batch with the DTM SUBMIT command or you execute the collection interactively with the DTM RUN command. The following DTM SUBMIT command invokes both DEC/Test Manager and the VAX Performance and Coverage Analyzer in batch:

```
DTM> SUBMIT PCS_COLLECTION /NOTIFY
%DTM-S-SUBMITTED, collection PCS_COLLECTION submitted
-DTM-I-TEXT, Job LM_COLLECT_1 (queue CLUSTER_BATCH, entry 314) started on ONE_BATCH
DTM>

Job LM_COLLECT_1 (queue ONE_BATCH, entry 314) completed
DTM>
```

When you create a collection, DEC/Test Manager builds a test collection command file which invokes the test template files and all related files for all tests you specify. If you use DEC/Test Manager with the VAX Performance and Coverage Analyzer, the collection command file performs the actions described in the following sections when you execute the collection. This is also shown in Figure 7-1.

The collection command file first defines all the global variables, DTM\$COLLECTION_NAME and PCA\$DATAFILE. The DTM\$COLLECTION_NAME variable is the global symbol identifying the collection. The PCA\$DATAFILE variable is the VAX Performance and Coverage Analyzer logical name for the data file to contain the results of the Collector run. DEC/Test Manager uses the logical name PCA\$DATAFILE to pass the name of the data file and its location to the VAX Performance and Coverage Analyzer.

The collection command file then runs the collection prologue file. This informs the VAX Performance and Coverage Analyzer of the location of the Collector initialization file and performs any setup tasks required for all the tests.

Then, for all tests, the collection command file runs each test as it is presented in its test description file. For each test, the collection command file performs the following functions:

- Defines all local variables and defines overrides for global variables.
- Defines DTM\$TEST_NAME, the local symbol identifying the test that is being run.

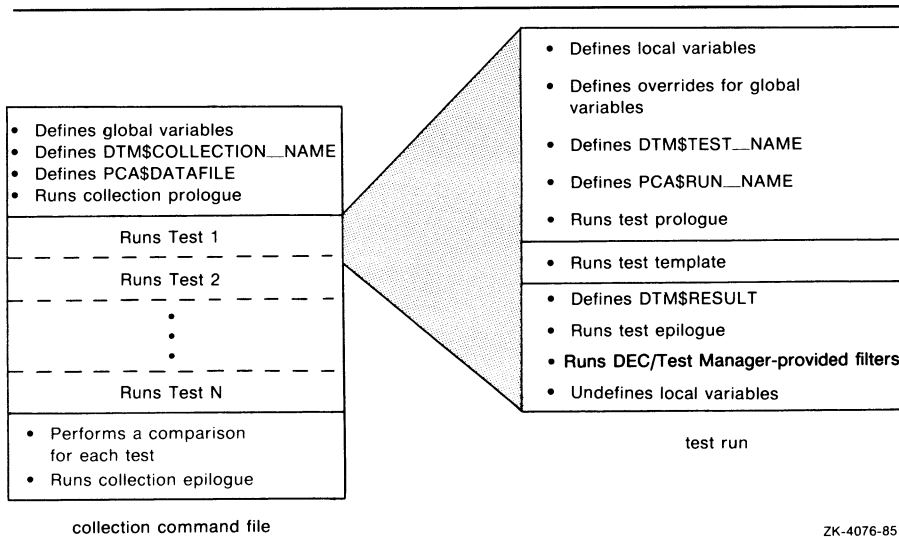
- Defines `PCA$RUN_NAME`, the VAX Performance and Coverage Analyzer logical name for the current Collector run. It is the same as `DTM$TEST_NAME`.
- Runs the test prologue file.
- Runs the test template, invokes the executable image of the program being tested, and gathers the performance and coverage data.
- Defines `DTM$RESULT`, the logical name for the file containing the test results.
- Runs the test epilogue file.
- Undefines local variables.

After all tests are run, the collection command file compares the result file created for each test with the benchmark file for that test. The collection command file then runs the collection epilogue file.

DEC/Test Manager provides **variables** that are symbols or logical names that you create to tailor your DEC/Test Manager environment. When you create a variable, you provide DEC/Test Manager with a DCL symbol or logical name that you want defined when running a test. For more information about variables, see Chapter 6.

Variables are the main run-time communication mechanism between DEC/Test Manager and the VAX Performance and Coverage Analyzer. Using the logical name `PCA$DATAFILE`, DEC/Test Manager passes the name of the data file containing data the Collector gathers to the VAX Performance and Coverage Analyzer. Using the logical name `PCA$RUN_NAME`, DEC/Test Manager passes the name of the current test for the current Collector run to the VAX Performance and Coverage Analyzer.

Figure 7-1: Phases in Test Execution When Using DEC/Test Manager with the VAX Performance and Coverage Analyzer



7.2.6 Using the Analyzer from the Review Subsystem

After DEC/Test Manager finishes running the collection, you can review the test results and examine the performance and coverage data. If you specify your own data file name in the Collector initialization file, invoke the Analyzer at DCL level and enter the data file name you specified. This process is described in the *VAX Performance and Coverage Analyzer User's Reference Manual*.

However, if you let DEC/Test Manager specify the name of the data file by default, then you should use the DEC/Test Manager Review subsystem PCA command to examine the performance and coverage data gathered from your program during the execution of your test system.

The PCA command, shown in the following example, invokes the Analyzer from within the Review subsystem. This command accepts no qualifiers or parameters and is described in the DEC/Test Manager Command Dictionary.

DTM_REVIEW> PCA

VAX Performance and Coverage Analyzer Version 1.1

PCAA>

You must be in the DEC/Test Manager Review subsystem when you issue the PCA command. You also must be positioned at a DEC/Test Manager result description to issue this command. (The positioning commands are FIRST, LAST, NEXT, BACK, and SELECT.) When you enter the PCA command, DEC/Test Manager spawns a subprocess to invoke the Analyzer. The command line that spawns the Analyzer as a subprocess also specifies that the Collector data file created during the batch run of your tests be used as input to the Analyzer.

DEC/Test Manager sets up the Analyzer filter DTM_FILTER to include only the data that was gathered when the current test (that is, the test at which you are now positioned) was run. In this way, the Analyzer can examine the data the Collector gathers on a test by test basis.

If you want to examine the data the Collector gathers as averaged over all tests in your test system rather than on a test by test basis, you can cancel DTM_FILTER (the Analyzer filter) by issuing the Analyzer CANCEL FILTER command:

PCAA> CANCEL FILTER DTM_FILTER

If you want to examine data for a particular test in your DEC/Test Manager collection and that test is not the current test, you can use the Analyzer SET FILTER command to redefine the filter so that it specifies the name of the test whose results you want to examine next:

PCAA> SET FILTER DTM_FILTER RUN_NAME = testname

In this example, testname is the result-description-name of the test whose data you want to examine next.

If you want DEC/Test Manager to filter the tests for you so that you can examine data separately for each test in your DEC/Test Manager collection, you should perform the following steps:

1. Exit the Analyzer to return to the Review prompt DTM_REVIEW> .
2. Position yourself at a different test using the DEC/Test Manager Review commands for positioning: FIRST, LAST, NEXT, BACK, and SELECT.
3. Enter the PCA command to use the Analyzer on the data for the newly selected test.

These three steps are more complex than canceling the filter or changing the filter definition, but they do automate the selection of tests and allow you to review test data for any number of your tests on a test by test basis.

7.3 A Sample Session Using These Two Tools Together

To use DEC/Test Manager with the VAX Performance and Coverage Analyzer, you must first link the program you want tested (RUNOFF.FOR) with the Collector's shareable image:

```
⌘ FORTRAN/DEBUG RUNOFF.FOR
⌘ LINK/DEBUG=SYS$LIBRARY:PCA$OBJ.OBJ RUNOFF.OBJ
⌘
```

Because RUNOFF.FOR has now been linked with the Collector, when DEC/Test Manager executes RUNOFF in batch the Collector is invoked in batch, and data is gathered on RUNOFF when tests are run on it.

Now create a Collector initialization file that contains the commands you want passed to the Collector. This file must contain a GO command; it may optionally contain other Collector commands. GO must always be the last command issued. The following example enables collection of system service counts and measurement of test coverage by codepath over the entire program. In addition, output verification has been selected, the collection of PC values from the VAX Call Stack has been selected, and the file PCA_DTM.LOG has been specified as the output of the logging session.

```
! Test coverage Collector initialization file
!
SET VERIFY                               !turn on output verification
SET LOG PCA_DTM.LOG                       !turn on output logging
SET SERVICES                              !gather system service counts
SET COVERAGE PROGRAM_ADDRESS BY CODEPATH !gather coverage by codepath
SET STACK_PCS                             !gather data from Call Stack
GO                                         !begin collection
```

Now that you have supplied all the information the Collector requires to run in batch, create a DEC/Test Manager collection prologue file that specifies the Collector initialization file for the VAX Performance and Coverage Analyzer to use. You can optionally use the collection prologue file to perform other tasks.

The collection prologue file in the following example defines the Collector initialization file to be MYINITFILE.PCAC and then redefines the logical name for the program's executable image to be the image that has been linked with the Collector.

```
#!      Collection prologue file for running the Collector in batch mode
#!
#!      define the Collector initialization file
#!
$ DEFINE PCAC$INIT DRA1:[PUBLIC]MYINITFILE.PCAC
#!
#!      define the Collector-linked exe
#!
$ DEFINE RUNOFF$EXE DRA1:[PUBLIC]PCA_RUNOFF.EXE
#!
#!      End of collection prologue file
$
```

By redefining the logical name in the collection prologue file, you can use the /PROLOGUE qualifier with the CREATE COLLECTION command as a single switch between a collection run without the VAX Performance and Coverage Analyzer and that same collection run with the VAX Performance and Coverage Analyzer.

You have now completed all the steps necessary to use these two tools together.

Now create a collection using DEC/Test Manager. (Note that, in the following example, the /PROLOGUE qualifier specifies the collection prologue file from the previous example.) In this example, the asterisk (*) includes in the collection MARGINS_COLLECT_1 all the tests in the DEC/Test Manager library. Note that the asterisk (*) is the wildcard character for test descriptions.

After creating the collection, execute it. Then, after the tests have run, enter the Review subsystem to review the test results and to use the Analyzer.

```
DTM> CREATE COLLECTION MARGINS_COLLECT_1 * -
_DTM> /PROLOGUE=DRA1:[PUBLIC]PCA_PROLOGUE
%DTM-S-CREATED, collection MARGINS_COLLECT_1 created
DTM> SUBMIT MARGINS_COLLECT_1 /NOTIFY
%DTM-S-SUBMITTED, collection MARGINS_COLLECT_1 submitted
-DTM-I-TEXT, Job MARGINS_COLLECT_1 (queue CLUSTER_BATCH, entry 314) started
on ONE_BATCH
DTM>

Job MARGINS_COLLECT_1 (queue ONE_BATCH, entry 314) completed
DTM>
```

Invoke the Review subsystem, select a specific test, and use the PCA command to invoke the Analyzer. After invoking the Analyzer, you can display the performance and coverage data gathered during the batch job for the selected test. In the following example, the test selected is the first test. See the *VAX Performance and Coverage Analyzer User's Reference Manual* for instructions on using the Analyzer.

```
DTM> REVIEW MARGINS_COLLECT_1
Collection MARGINS_COLLECT_1 with 1 test was created on 19-MAR-1986 15:08:55
by the command:
  CREATE COLLECTION MARGINS_COLLECT_1 * /PROLOGUE=DRA1:[PUBLIC]PCA_PROLOGUE
  Last Review Date = not previously reviewed
  Success count = 0
  Unsuccessful count = 1
  New test count = 0
  Updated test count = 0
  Comparisons aborted = 0
  Test not run count = 0
DTM_REVIEW> FIRST
Result Description LM_TEST_1                Comparison Status : Unsuccessful
DTM_REVIEW> PCA

      VAX Performance and Coverage Analyzer Version 1.1

PCAA>
```

7.4 Notes of Interest

This section describes information you should read and understand before you use DEC/Test Manager and the VAX Performance and Coverage Analyzer together.

7.4.1 System Considerations When Using These Two Tools Together

Both DEC/Test Manager and the VAX Performance and Coverage Analyzer can generate large files and large numbers of files. It is best to make sure that you have sufficient system resources before you begin creating the files that result from each of these tools.

7.4.2 Using the Collector SET COVERAGE Command with the /PREVIOUS Qualifier

The VAX Performance and Coverage Analyzer provides a way to improve running speed by reducing the size of the data file created when collecting coverage information on your program. When you use the /PREVIOUS qualifier with the Collector SET COVERAGE command, the VAX Performance and Coverage Analyzer records only a single record for each breakpoint hit during an entire run of a DEC/Test Manager collection. When you later analyze the collected data, information appears more quickly on the screen, but coverage data is accurate *only* for the first test in the collection run.

If you use the /PREVIOUS qualifier when collecting coverage data, use the Analyzer CANCEL FILTER command and examine coverage for the entire DEC/Test Manager collection as a unit.

```
PCAA> CANCEL FILTER DTM_FILTER
```

NOTE

If you want to examine coverage data for individual tests, do not use the /PREVIOUS qualifier on the Collector SET COVERAGE command.



Command Dictionary

The first section of the Command Dictionary describes the elements of the DEC/Test Manager command line and defines the syntax rules for entering commands.

The last section of the Command Dictionary describes each DEC/Test Manager command. The commands are arranged in alphabetical order with each command description containing the following:

- General format for the command
- List of command qualifiers
- Restrictions on the use of the command
- List of parameter qualifiers, if applicable
- Descriptions of all command parameters
- Explanation of the command
- Descriptions of all command qualifiers
- Descriptions of any parameter qualifiers
- Examples



CD.1 Command Format

A DEC/Test Manager command line consists of the command verb DTM and a combination of the following elements:

- A one- or two-word command-name (required)
- One or more command qualifiers
- One or more parameters
- A remark

DEC/Test Manager commands have the following format:

```
DTM command-name [/qualifier(s)] [parameter(s)] [remark]
```

The **command-name** describes the DEC/Test Manager action. **Command qualifiers** and **command parameters** further specify the action of the command. You can place command qualifiers anywhere after the command-name. When you specify the test-group-expression parameter with **parameter qualifiers**, each parameter qualifier you use must immediately follow the item in the test-group-expression to which it refers.

The **remark** is an optional parameter on all commands which modify the DEC/Test Manager library in any way. The remark allows you to associate a comment with the command.

The proper formats for the DEC/Test Manager commands, command parameters, command qualifiers, and parameter qualifiers are discussed in the following sections.

CD.2 Command Parameters

The following is a list of the command parameters used with DEC/Test Manager commands. The syntax for each parameter is described in the indicated section.

- Collection-name (Section CD.2.1)
- File-spec (Section CD.2.2)
 - Directory-spec (Section CD.2.2.1)
 - File-name (Section CD.2.2.2)
 - File-type and -version (Section CD.2.2.3)

- Group-name (Section CD.2.3)
- Remark (Section CD.2.4)
- Result-description-name (Section CD.2.5)
- Test-name (Section CD.2.6)
- Variable-name (Section CD.2.7)
- Variable-value (Section CD.2.8)
- Expressions (Section CD.2.9)
 - Collection-expression (Section CD.2.9.1)
 - Group-expression (Section CD.2.9.2)
 - Object-expression (Section CD.2.9.3)
 - Result-description-expression (Section CD.2.9.4)
 - Test-expression (Section CD.2.9.5)
 - Test-group-expression (Section CD.2.9.6)
 - Variable-expression (Section CD.2.9.7)
- Wildcard Characters (Section CD.2.10)

CD.2.1 Collection-Name

A **collection-name** is a name that uniquely identifies a set of test descriptions as a collection. Collections are used in DEC/Test Manager as a way of building a set of tests you want to run together. Follow the same rules as for file-names in creating collection-names. These rules are described in Section CD.2.2.2.

You can use the same name for a collection, group, test description, or variable. A collection-name cannot begin with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

CD.2.2 File-Spec

A **file-spec** is a standard VAX/VMS file specification, including device, directory, file-name, file type, and version number. The format is

```
device:[directory]filename.type;version
```

You cannot include a node specification on any file specification while using DEC/Test Manager.

CD.2.2.1 Directory-Spec

A **directory-spec** is a valid VAX/VMS directory specification. The device name cannot exceed 15 characters, and the directory name cannot exceed 255 characters.

The format is

`device: [directory]`

Logical names also can be used to specify this parameter.

CD.2.2.2 File-Name

A **file-name** is the file-name component of a VAX/VMS file specification. This is the portion of a file specification that precedes the period (.) and file type. A file-name can be from 1 to 39 characters long. It can consist of any or all of the following:

- Alphanumeric characters (a through z, A through Z, 0 through 9)
- A dollar sign (\$)
- An underscore (_)
- The subset of the DEC Multinational Character Set described in the *VAX/VMS DCL Dictionary*

A file-name must begin with an alphanumeric character.

These rules are also followed for creating all the DEC/Test Manager parameters that specify names, for example, test-names and collection-names.

CD.2.2.3 File-Type and Version

A file-type and version are components of a VAX/VMS file specification. The file-type is the portion of a file specification that follows the period (.) and file-name. If a file-type is specified, it must be separated from the file-name with a period. A file-type can be from 1 to 39 characters long. The version follows the file-type and is separated from it by a semicolon or by another period. It is a decimal number designating the version of the file. Do not include a version number when specifying files to DEC/Test Manager. For more information about file-types and versions, see the *VAX/VMS DCL Dictionary*.

CD.2.3 Group-Name

A **group-name** is a name that uniquely identifies a set of test descriptions as a group. Groups are used in DEC/Test Manager as a means of categorizing test descriptions, subgroups of test descriptions, or both. Follow the same rules as for file-names in creating group-names.

You can use the same name for a collection, group, test description, and variable. You cannot use a group-name beginning with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

CD.2.4 Remark

A **remark** is a string that functions as a comment. A remark can consist of any printable ASCII characters and is limited in length so that the total command-line length follows VAX/VMS conventions. If a remark includes any space characters, it must be enclosed in double quotation marks ("").

This is an optional parameter on all commands that modify the DEC/Test Manager library. The remark is stored in the DEC/Test Manager library history file which is displayed when you issue the SHOW HISTORY command.

The MODIFY GROUP, COPY TEST_DESCRIPTION, MODIFY TEST_DESCRIPTION, and MODIFY VARIABLE commands all take both the optional remark parameter and an optional remark qualifier. The remark parameter specifies the remark logged with the COPY or MODIFY command in the history file. The remark qualifier specifies the remark associated with the modified or copied group, test description, or variable.

Note also that the REMARK command allows you to enter a remark directly into the history file to log an unusual event in the DEC/Test Manager library.

CD.2.5 Result-Description-Name

A **result-description-name** is a name that uniquely identifies a result description. The result-description-name for a test is created by DEC/Test Manager and is the same as the test-name. The result-description-name is used to identify a specific result description when you are using the Review subsystem.

CD.2.6 Test-Name

A **test-name** is a name that uniquely identifies a test description. A test description contains all the information DEC/Test Manager uses to run a test. Follow the same rules as for file-names in creating test-names.

You can use the same name for a collection, group, test description, or variable. You cannot use a test-name beginning with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

CD.2.7 Variable-Name

A **variable-name** is a name that uniquely identifies a variable. A variable is a user-defined VAX/VMS symbol or logical name stored in the DEC/Test Manager library. Follow the same rules as for file-names in creating variable-names.

You can use the same name for a collection, group, test description, or variable. You cannot use the variable-names P1 through P8, nor can you use variable-names beginning with DTM\$. These names are reserved for use by DEC/Test Manager.

CD.2.8 Variable-Value

A **variable-value** is the value assigned to a variable. A variable-value can consist of all printable ASCII characters. If a variable-value includes any space characters, it must be enclosed in double quotation marks (""). Follow the same rules as for the VAX/VMS restriction on total command line length.

Be sure to use an equal sign (=) when assigning variable values with the CREATE and MODIFY TEST_DESCRIPTION commands and with the CREATE COLLECTION command. DEC/Test Manager does not accept the colon (:) as an alternative to the equal sign in this instance.

CD.2.9 Expressions

An **expression** lets you name multiple instances of the same parameter in a single parameter field. An expression can be composed of one or more of the following:

- A name
- A wildcard character
- A wildcard character used in combination with a name or partial name
- A list of the preceding items with the items separated by commas

For example, LMTEST* in a test-expression field is equivalent to the names of all test descriptions in the library that begin with LMTEST.

NOTE

The result-description-expression is the one exception to these rules. A result-description-expression can consist of a result-description-name, a wildcard character, or a full or partial result-description-name in combination with wildcard characters. It cannot consist of a list of these items separated by commas.

The six types of expressions in DEC/Test Manager are as follows:

- Collection-expression
- Group-expression
- Object-expression
- Result-description-expression
- Test-expression
- Test-group-expression
- Variable-expression

Collection-expressions, group-expressions, result-description-expressions, test-expressions, and variable-expressions all contain names of the same type. For example, a collection-expression contains only collection-names and expressions that resolve to collection-names.

Object-expressions and test-group-expressions contain more than one type of name. Object-expressions can contain test-names, group-names, and collection-names. Test-group-expressions can contain only test-names, group-names, and wildcard expressions that resolve to these names.

CD.2.9.1 Collection-Expression

A **collection-expression** identifies one or more collections. When you use a collection-expression, follow the rules presented in Section CD.2.9, where the names you specify are collection-names.

CD.2.9.2 Group-Expression

A **group-expression** identifies one or more groups. When you use a group-expression, follow the rules presented in Section CD.2.9, where the names you specify are group-names.

CD.2.9.3 Object-Expression

An **object-expression** identifies one or more tests, groups of tests, or collections of tests. When you use an object-expression, follow the rules presented in Section CD.2.9, where the names you specify are test-names, group-names, or collection-names.

Object-expressions are used with the SHOW HISTORY command to identify the test, group, or collection about which you want information.

CD.2.9.4 Result-Description-Expression

A **result-description-expression** identifies one or more result descriptions. When you use a result-description-expression, follow the first three rules presented in Section CD.2.9, where the names you specify are result-description-names. A result-description-expression can consist of a result-description-name, a wildcard character, or a full or partial result-description-name in combination with wildcard characters. It cannot consist of a list of these items separated by commas.

CD.2.9.5 Test-Expression

A **test-expression** identifies one or more tests. When you use a test-expression, follow the rules presented in Section CD.2.9, where the names you specify are test-names.

CD.2.9.6 Test-Group-Expression

A **test-group-expression** identifies one or more tests or groups of tests. When you use a test-group-expression, follow the rules presented in Section CD.2.9, where the names you specify are test-names and group-names.

The items in a test-group-expression are identified with the parameter qualifiers /GROUP and /TEST_DESCRIPTION. DEC/Test Manager checks after each item in the test-group-expression to see whether it is a group item or a test item. The /GROUP qualifier immediately following an item in a test-group-expression identifies that item as a group-name. Similarly, the /TEST_DESCRIPTION qualifier immediately following an item in a test-group-expression identifies that item as a test-name. If an item is not followed by a qualifier, DEC/Test Manager assumes that the item identifies a test-name.

For example, the test-group-expression SUB/GROUP,SUB1 specifies the group SUB and the test SUB1 (SUB is explicitly identified with the /GROUP qualifier; SUB1, lacking the qualifier, is identified as a test-name by default).

Test-group-expressions are used with the CREATE COLLECTION command and with the INSERT, MODIFY, REMOVE, and SHOW TEST_DESCRIPTION commands.

CD.2.9.7 Variable-Expression

A **variable-expression** identifies one or more variables. When you create a variable-expression, follow the rules presented in Section CD.2.9, where the names you specify are variable-names.

CD.2.10 Wildcard Characters

The **wildcard characters** are the asterisk (*) and percent sign (%). Use asterisks for partial- or full-field substitutions and percent signs for single-character substitutions. Follow the same rules as for the VAX/VMS operating system when using wildcards.

CD.3 Qualifiers

The two types of DEC/Test Manager qualifiers are command qualifiers and parameter qualifiers. Both types of qualifiers and their uses are described in the following sections.

CD.3.1 Command Qualifiers

The DEC/Test Manager command qualifiers modify the command and can be placed anywhere on the command line after the command-name. You can enter command qualifiers before or after any parameters and before or after the remark. Some command qualifiers require a value.

The following qualifiers can be used with most DEC/Test Manager commands. The **/CONFIRM** qualifier is the default on all **DELETE** and **REMOVE** commands. The **/LOG** qualifier is the default wherever it occurs. Unless otherwise specified, the **/OUTPUT** qualifier defaults to **SYS\$OUTPUT**.

/CONFIRM
/NOCONFIRM

Controls whether the name of each item being processed is displayed before processing occurs. You are asked to confirm whether the process should proceed. You can reply with one of the following:

- Y or YES to proceed for that item
- N or NO to halt the process for that item
- A or ALL to proceed for all items without further confirmation
- Q or QUIT to halt the processing for all items

If input is being taken from a command file or if the command is issued in batch, the action is performed without confirmation—regardless of whether the **/CONFIRM** or **/NOCONFIRM** qualifier is specified.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

/OUTPUT[=file-spec]

Sends information to the default output device **SYS\$OUTPUT**, or to the specified file.

CD.3.2 Parameter Qualifiers

The two parameter qualifiers are /GROUP and /TEST_DESCRIPTION.

Parameter qualifiers are used only within the test-group-expression parameter to differentiate the items contained in the test-group-expression as representing either tests or groups.

The position of these qualifiers is significant. The /GROUP parameter qualifier identifies as a group-name the item that it follows. Similarly, the /TEST_DESCRIPTION parameter qualifier identifies as a test-name the item that it follows. The default parameter qualifier is /TEST_DESCRIPTION. Thus, if no parameter qualifier follows an item in a test-group-expression, DEC/Test Manager assumes that item is a test-name. For example:

```
LMTEST1, LMTESTS/GROUP, RMTEST
```

In this test-group-expression, DEC/Test Manager identifies LMTEST1 and RMTEST as test-names and LMTESTS as a group-name.

CD.4 Rules for Entering and Canceling Commands

This section describes guidelines to follow when using DEC/Test Manager commands.

CD.4.1 Entering Commands

At the DCL prompt, you can enter DEC/Test Manager commands by beginning each command with DTM, or you can enter DTM and press RETURN to enter the DEC/Test Manager subsystem where you receive the DTM> prompt. You must omit the letters DTM when you enter commands after this prompt. Using the DEC/Test Manager subsystem is faster and more convenient when you issue many consecutive commands, because you activate the DEC/Test Manager image only once.

If you press RETURN before completing a command, you are prompted for all required information for a command except required qualifiers. You can type a command string on more than one line by using the DCL continuation character, a hyphen (-), as the last character on a command line. DEC/Test Manager commands follow VAX/VMS conventions for line length.

When you invoke DEC/Test Manager in subsystem mode from a command procedure, be sure to supply all command parameters. When you omit a parameter for which DEC/Test Manager would prompt you if you entered a command interactively, DEC/Test Manager reads the next line in the command procedure as the missing parameter rather than as a separate command, and the command is lost. This applies to both DEC/Test Manager and DCL command procedures.

CD.4.2 Abbreviating Commands

All DEC/Test Manager commands and qualifiers can be abbreviated to their minimum unique form (usually two or three characters). If you use an ambiguous abbreviation, the command is not processed and DEC/Test Manager issues an error message.

CD.4.3 Canceling Commands

If you wish to cancel a wildcard transaction before it has completed, press CTRL/C. This action can be useful, for example, to cancel a SHOW command after you have obtained the desired information. If you press CTRL/C during a wildcard transaction that updates the library, DEC/Test Manager finishes the current transaction, but does not continue.

When you enter a wildcard command from the DCL-prompt level and then press CTRL/C during execution of the command, DEC/Test Manager returns control to DCL (as indicated by the dollar sign (\$) prompt). If you enter the command from the DEC/Test Manager subsystem prompt level, control is returned to DEC/Test Manager as indicated by the DEC/Test Manager DTM> prompt. You can also press CTRL/C at any command prompt to abort the current prompt level or the Review subsystem.

CD.4.4 Leaving DEC/Test Manager

You can type EXIT or press CTRL/Z to exit from DEC/Test Manager. If used at the DTM-prompt level, typing EXIT or pressing CTRL/Z returns you to the DCL-prompt level. If used at the DTM-Review level, typing EXIT or pressing CTRL/Z returns you to the DTM-prompt level, if you were in DTM-prompt mode when you initiated the Review session, or to DCL-prompt level if you were not.

CD.5 Command Execution

Most DEC/Test Manager commands take place in your main process. Unlike these commands, the following commands take place in a subprocess of your main process:

```
PLAY
RUN
CREATE TEST_DESCRIPTION/RECORD
CREATE TEST_DESCRIPTION/INPUT
MODIFY TEST_DESCRIPTION/RECORD
MODIFY TEST_DESCRIPTION/INPUT
```

When you issue one of these four commands, DEC/Test Manager spawns a subprocess where the command takes place and creates a pseudo terminal which handles all input and output for the subprocess.

The PLAY and RUN commands take input from the specified SESSION files and pass it on to the pseudo terminal. The application being tested takes input from the pseudo terminal as if you had entered the input at the pseudo terminal. Both commands send output to the pseudo terminal. The RUN command also sends output to the result and benchmark files for each test in the collection. Note that when you execute a collection with the RUN command, the prologue and epilogue files for the collection and for each test execute in your main process, not in the subprocess.

The CREATE and MODIFY TEST_DESCRIPTION/RECORD commands take input from your terminal and pass it to both the SESSION file and to the pseudo terminal.

Command Descriptions



(\$) DTM

Invokes DEC/Test Manager as a subsystem. The DTM command is entered at the DCL prompt (\$).

Format

DTM

Command Qualifier

/[NO]INIT

Default

/INIT

Command Qualifier

/INIT

/NOINIT

Specifies whether DTM executes the initialization file (defined by the logical DTM\$INIT) when invoked.

The default is to execute any existing initialization file whenever DEC/Test Manager is invoked.

Description

The DTM command invokes DEC/Test Manager in subsystem mode and displays the DTM> prompt. The DTM command must be entered at the DCL prompt. Leave the DEC/Test Manager subsystem by typing the EXIT command or by pressing CTRL/Z.

You can also use the DEC/Test Manager subsystem from batch or command files; both give you the same overhead savings. However, you must be sure that all required parameters are on each command line. If a parameter is missing, DEC/Test Manager takes the next line as the missing parameter.

(\$) DTM

Example

```
⌘ DTM  
DTM>
```

This command invokes DEC/Test Manager as a subsystem.

DTM @file-spec

Executes DEC/Test Manager commands contained in the specified file.

Format **DTM @file-spec**

Command Qualifiers	Defaults
None.	None.

Command Parameter

file-spec

Specifies the command procedure to execute. If the file-spec does not include a file-type, DEC/Test Manager assumes the default file-type COM.

Description

The @file-spec command executes the commands in the specified file. The file can contain any DEC/Test Manager command, including another @file-spec command. Do not preface the commands with DTM or \$. For example, enter SHOW LIBRARY, not DTM SHOW LIBRARY.

When DEC/Test Manager executes an EXIT command or reaches the end of the command procedure, it leaves you at the current command level. The invoking command stream can be either the terminal or another command procedure.

Example

```
DTM> @SUBMIT_LMTEST1
%DTM-S-LIBIS, DEC/Test Manager library is DRA1:[PROJECT.MYLIB]
%DTM-S-RESUBMITTED, collection LMTEST1 has been resubmitted
-DTM-I-TEXT, Job LMTEST1 (queue SYS$BATCH, entry 1862) started on SYS$BATCH
DTM>
```

This example executes the command procedure SUBMIT_LMTEST1.COM containing the commands SET LIBRARY DRA1:[project.mylib] and SUBMIT lctest1.

DTM ATTACH

DTM ATTACH

Switches control from your current process to another process in your job.

Format **DTM ATTACH** [*process-name*]

Command Qualifiers

/IDENTIFICATION=pid
/PARENT

Defaults

None.
None.

Command Qualifiers

/IDENTIFICATION=pid

Specifies the process identification (PID) of the process to which you want to attach your terminal. You can omit the leading zeros when you specify a PID.

If you specify the */IDENTIFICATION* qualifier, do not specify the process-name parameter or the */PARENT* qualifier. If you do not specify a qualifier, you must specify a process-name.

/PARENT

Specifies that the process you want to attach to is your original, or parent, process.

If you specify the */PARENT* qualifier, do not specify the process-name parameter or the */IDENTIFICATION* qualifier. If you do not specify a qualifier, you must specify a process-name.

Command Parameter

process-name

Specifies an existing process to which you want to attach your terminal.

If you specify either the */IDENTIFICATION* or */PARENT* qualifier, do not specify the process-name parameter or the other qualifier. If you do not specify a qualifier, you must specify a process-name.

Description

The ATTACH command allows you to connect your terminal to another process in your job. You can use the ATTACH command to change control between subprocesses you have created with the SPAWN command or to reconnect to your parent process. You can also use the ATTACH command in conjunction with the SPAWN/WAIT command to return to your DEC/Test Manager session without terminating the subprocess. See the SPAWN command for more information.

Example

```
MAIL> SPAWN DTM
DTM> ATTACH/PARENT
You have 0 new messages.
```

```
MAIL>
```

This example uses the MAIL SPAWN command to create a subprocess running DEC/Test Manager. The DEC/Test Manager ATTACH command is then used to attach your terminal back to the MAIL session, your parent process.

DTM COMPARE

DTM COMPARE

Compares the result file produced for each test description in a collection with its benchmark file. This command works only with a collection that has been run but not reviewed.

Format **DTM COMPARE** *collection-name*

Command Qualifiers

/CHARACTERS
/FULL
/IGNORE=(keyword,...)
/[NO]LOG
/[NO]PARALLEL
/RECORDS
/SCREENS
/WIDTH=n

Defaults

See text.
None.
None.
/LOG
/NOPARALLEL
/RECORDS (for noninteractive tests)
/SCREENS (for interactive tests)
/WIDTH=132

Command Qualifiers

/CHARACTERS

Performs a character by character comparison of the results file with the benchmark file.

The default is **/SCREENS** for interactive tests and **/RECORDS** for noninteractive tests.

/FULL

Includes in the difference file a complete listing of the text that was identical and the differences encountered when the result file and benchmark file were compared.

/IGNORE=(keyword,...)

Indicates one or more types of special characters to be ignored when the result and benchmark files are compared. The following list contains the available keywords.

DTM COMPARE

If you specify the `/IGNORE` and `/SCREENS` qualifiers together, DEC/Test Manager performs the comparison screen by screen and ignores the `/IGNORE` qualifier.

Keyword	Ignored Characters
<code>CASE</code>	Ignores any differences between the case of alphabetic characters (A through Z, a through z) as it compares the two files.
<code>FORM_FEEDS</code>	Ignores form-feed characters as it compares the two files.
<code>LEADING_BLANKS</code>	Ignores leading blanks and tabs as it compares the two files.
<code>TRAILING_BLANKS</code>	Ignores trailing blanks and tabs as it compares the two files.
<code>SPACING</code>	Treats multiple blanks and tabs as a single space as it compares the two files.

If you specify more than one keyword, separate the keywords with commas and enclose the list in parentheses. The output file (your result file) is not changed in any way by the `/IGNORE` qualifier.

`/LOG`

`/NOLOG`

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is `/LOG`.

`/PARALLEL`

`/NOPARALLEL`

Specifies whether the lines that do not match in the result and benchmark files are formatted side by side.

If you specify the `/[NO]PARALLEL` and `/SCREENS` qualifiers together, DEC/Test Manager performs the comparison screen by screen and ignores the `/[NO]PARALLEL` qualifier.

The default is `/NOPARALLEL`.

DTM COMPARE

/RECORDS

Specifies that the comparison of the result and benchmark files is to be performed on a record by record basis. Records are identical only if they contain the same characters. Use this type of comparison only when you expect the record in which a string appears to be the same each time a comparison is performed.

Use caution when specifying the */RECORDS* qualifier for an interactive test because the records in the result file are not guaranteed to be written the same way each time the test runs. You might want to use the */RECORDS* qualifier for an interactive test whose result file is not generated by DEC/Test Manager, for example, if you rename a test output file to be DTM\$RESULT.

The default is */SCREENS* for interactive tests and */RECORDS* for noninteractive tests.

/SCREENS

Performs a screen by screen comparison of the result and benchmark files for an interactive test.

The default is */SCREENS* for interactive tests and */RECORDS* for noninteractive tests.

/WIDTH=n

Specifies the maximum width allowed for the differences report. The minimum width is 48 columns and the maximum width is 511 columns.

The default value is 132 columns.

Command Parameter

collection-name

Specifies a name that uniquely identifies a collection of test descriptions to be compared. A collection-name can be from 1 to 39 characters long. You must specify a collection-name. You cannot enter a collection-expression containing wildcard characters.

Description

The COMPARE command prepares a collection to be reviewed. For each test description in a collection, DEC/Test Manager compares the results generated when the test was run with the test's benchmark file. If there are differences between the files, a difference file is generated, the result file is deleted, and the comparison status for the test is unsuccessful. If there are no differences, the test has the comparison status successful. If there is no benchmark file associated with the test, the comparison status for the test is new test. You can create a benchmark file for this test while reviewing the test results from the Review subsystem with the Review subsystem UPDATE command.

In addition to successful, unsuccessful, and new test, a test can have the comparison aborted, not run, or updated comparison status. Comparison status is discussed in Chapter 5.

If you store your benchmark files in a DEC/CMS library, DEC/Test Manager looks there for your benchmark files. If DEC/Test Manager finds that a result file has not been generated for a test, it marks the test as not run.

The COMPARE command can be used only with a collection that has executed but has not been reviewed. DEC/Test Manager informs you if you use the COMPARE command with a collection that has already been compared. By default, DEC/Test Manager automatically compares collections when they are executed. The only way to prevent a collection from being automatically compared is to specify the /NOCOMPARE qualifier on the CREATE COLLECTION command. You cannot use the COMPARE command with a collection that is in use.

The COMPARE command compares the completed part of a partially run collection. If you have a collection that does not run to completion, you can compare and review the tests in the collection that did run. A partially run collection results if the system crashes while the collection is executing, if you terminate the RUN command by pressing CTRL/C, or if you stop execution of the collection with the DEC/Test Manager STOP command.

The COMPARE command compares the results of both interactive and noninteractive tests. You can compare interactive tests in three ways. By specifying the /SCREENS qualifier, you compare your result file with the benchmark file on a screen by screen basis. By specifying the /CHARACTERS qualifier, you compare your result file with the benchmark file on a character by character basis. By specifying the /RECORDS

DTM COMPARE

qualifier, you compare your result file with your benchmark file on a record by record basis. The default for interactive tests is /SCREENS.

When you perform the comparison character by character, record boundaries in the result file are ignored. Line feed and escape characters are used to break the output into lines which are then processed for differences.

If an error occurs while a comparison is being performed, the test being compared is given the comparison aborted comparison status. The comparison status for a test is included in the test's result description.

Example

```
⌘ DTM COMPARE MARGIN_COLLECT_1
%DTM-I-SUCCEEDED, the comparison for the test LMTEST succeeded
%DTM-I-SUCCEEDED, the comparison for the test RMTEST succeeded
%DTM-S-COMPARED, collection MARGIN_COLLECT_1 compared
```

This example compares the results for all tests in the collection MARGIN_COLLECT_1. For each test, DEC/Test Manager deletes the result files for tests whose benchmark and result files match, and it creates a difference file for tests whose benchmark and result files differ.

DTM CONVERT LIBRARY

Converts DEC/Test Manager libraries created with a version of DEC/Test Manager prior to Version 2.0 for use with the current version of DEC/Test Manager.

Format	DTM CONVERT LIBRARY	<i>v1-library-name v2-library-name</i>
	Command Qualifiers	Defaults
	None.	None.

Command Parameters

v1-library-name

Specifies the directory for the existing DEC/Test Manager library you want to convert.

v2-library-name

Specifies the directory for the new DEC/Test Manager library you want to create.

Description

The CONVERT LIBRARY command creates a copy of an existing DEC/Test Manager library and converts the copy for use with this version of DEC/Test Manager. Libraries created with DEC/Test Manager Version 2.0 do not need to be converted. Conversion maintains everything in your existing library except collections. Your existing library is not altered.

To convert a library, first create an empty directory to contain the new, converted library.

DTM CONVERT LIBRARY

Example

```
⌘ CREATE/DIRECTORY [project.v2lib]
⌘DTM CONVERT LIBRARY [project.v1lib] [project.v2lib]
%DTM-S-COPIED, V1 variables copied
%DTM-S-COPIED, V1 groups copied
%DTM-S-COPIED, V1 test descriptions copied
-DTM-S-CONVERTED, your V1 library has been successfully converted to V2
⌘
```

This example first creates a new directory to contain the converted library. Then the existing library is converted. The benchmark files are not copied because they are stored outside the DEC/Test Manager library in a benchmark directory. They are accessible to the new library.

DTM COPY TEST_DESCRIPTION

/EPILOGUE=file-spec

/NOEPILOGUE

Determines whether a test epilogue file is associated with the test description.

The */EPILOGUE* qualifier replaces the existing epilogue file with the specified epilogue file.

The */NOEPILOGUE* qualifier specifies that no epilogue be associated with the new test description.

By default, the epilogue file associated with the existing test description is also associated with the new test description.

/NOFILTERS

Specifies that no filters be associated with the new test description.

By default, the filters associated with the existing test description are also associated with the new template file.

/NOGROUPS

Specifies that the new test description does not belong to any groups.

By default, the new test description belongs to the same groups as the old test description.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/PROLOGUE=file-spec

/NOPROLOGUE

Determines whether a test prologue file is associated with the test description.

The */PROLOGUE* qualifier replaces the existing prologue file with the specified prologue file.

The */NOPROLOGUE* qualifier specifies that no prologue be associated with the new test description.

By default, the prologue file associated with the existing test description is also associated with the new test description.

DTM COPY TEST_DESCRIPTION

/REMARK="string"

/NOREMARK

Determines whether a remark is associated with the new test description. Note that this remark is associated with the test description you are creating; it is not the remark logged with the COPY TEST_DESCRIPTION command.

The */REMARK* qualifier replaces the remark currently associated with the test description with the remark you specify.

The */NOREMARK* qualifier specifies that no remark be associated with the new test description.

By default, the remark associated with the new test description will be a copy of the remark associated with the existing test description.

/TEMPLATE=file-spec

/NOTEMPLATE

Specifies whether the existing template file is to be copied as is, or whether a default template file-name is to be supplied.

The */TEMPLATE* qualifier replaces the existing template file with the specified template file.

The */NOTEMPLATE* qualifier specifies a template file name of the form test-name.COM for a noninteractive test and test-name.SESSION for an interactive test.

By default, the template file associated with the existing test description is also associated with the new test description.

/NOVARIABLES

Specifies that no variables be associated with the new test description.

By default, the variables associated with the old test description are associated with the new test description.

Command Parameters

test-name1

Specifies the name of the test description to be copied—the existing test description. You cannot use wildcards to specify the test-name parameters. The test-name1 and test-name2 parameters must be different. You cannot copy a test description to itself.

DTM COPY TEST_DESCRIPTION

test-name2

Specifies the name of the test description to be created—the new test description. You cannot use wildcards to specify the test-name parameters. The test-name1 and test-name2 parameters must be different. You cannot copy a test description to itself.

remark

Specifies a string that contains a comment. This parameter is optional. If you do not provide a remark, you are prompted for one, but the test description is copied even if you do not provide a remark. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). This remark is associated with the COPY TEST_DESCRIPTION command and is logged with it in the history file.

Description

The COPY TEST_DESCRIPTION command makes a copy of an existing test-description. This allows you to create several similar test descriptions without entering information into all the test description fields. Because DEC/Test Manager does not permit you to have two test descriptions with the same name, the name for the new test description must be unique.

When a test description is copied, only the information in the fields you specify with command qualifiers is modified; information in the remaining test description fields is copied as is from the existing test description. By default, the new test description belongs to the same groups as the existing test description; it is associated with the same variables as the existing test description. If the test description you are copying describes an interactive test, the new test is marked interactive.

Example

```
⌘ DTM COPY TEST_DESCRIPTION LMTEST1 LEFTMARGIN /PROLOGUE=NEWPRO.COM
  _Remark:Left margin test with new prologue file
%DTM-I-DEFAULTED, benchmark file name defaulted to LEFTMARGIN.BMK
%DTM-S-COPIED, test description LMTEST1 copied
-DTM-S-CREATED, test description LEFTMARGIN created
⌘
```

This example creates a copy of the existing test description, LMTEST1, named LEFTMARGIN. The new test description is associated with the prologue named NEWPRO.COM.

DTM CREATE COLLECTION

Designates a set of tests as a collection.

Format **DTM CREATE COLLECTION** *collection-name*
test-group-expression
[remark]

Command Qualifiers

/[NO]BENCHMARK_DIRECTORY=dir-spec
/CLASS=(keyword=class-name,...)
/[NO]COMPARE[=(keyword,...)]
/[NO]EPILOGUE=file-spec
/[NO]LOG
/[NO]PROLOGUE=file-spec
/[NO]SUBMIT[=(keyword,...)]
/[NO]TEMPLATE_DIRECTORY=dir-spec
/VARIABLE=(variable-name=variable-value,...)
/[NO]VERIFY

Defaults

See text.
See text.
/COMPARE
Current collection epilogue.
/LOG
Current collection prologue.
/NOSUBMIT
See text.
See text.
/VERIFY

Parameter Qualifiers

/GROUP
/TEST_DESCRIPTION

Defaults

/TEST_DESCRIPTION
/TEST_DESCRIPTION

Command Qualifiers

/BENCHMARK_DIRECTORY=dir-spec
/NOBENCHMARK_DIRECTORY

Determines whether for this collection DEC/Test Manager should search the default benchmark directory for benchmark files. If for a test within the collection a directory is included in the benchmark file specification, DEC/Test Manager always searches that directory for the benchmark file. If a directory is not included, DEC/Test Manager searches the default directory for the benchmark file.

DTM CREATE COLLECTION

The `/BENCHMARK_DIRECTORY` qualifier overrides for this collection the default benchmark directory and DEC/Test Manager searches the specified directory. The directory you specify can be either another directory or a DEC/CMS library.

The `/NOBENCHMARK_DIRECTORY` qualifier overrides for this collection the default benchmark directory. DEC/Test Manager searches your default directory for all benchmark files without directory specifications.

By default, DEC/Test Manager searches the default benchmark directory.

`/CLASS=(keyword=class-name,...)`

Specifies the optional DEC/CMS class for benchmark files and template files stored in DEC/CMS libraries with the keywords `BENCHMARK` and `TEMPLATE`. The class names you specify can be the same, or you can use different class names for your benchmark and template files. If you specify both keywords, separate them with a comma and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses.

If you do not specify a class and the file is stored in a DEC/CMS library, the latest generation on the main line of descent is used.

`/COMPARE[=(keyword,...)]`

`/NOCOMPARE`

Determines whether DEC/Test Manager compares the results of each test with its benchmark file (the file that contains expected test results) after the collection is executed.

The `/COMPARE` qualifier specifies that DEC/Test Manager is to compare all tests after the collection is executed. A collection must be compared before it can be reviewed. Any differences between the results for a test and its benchmark file are recorded in a difference file for that test. Tests without benchmarks can be compared. When these tests are reviewed, benchmarks can be generated for them. When you enter the `/COMPARE` qualifier, by default the `COMPARE` command default qualifiers (`/SCREENS`, `/LOG`, and `/WIDTH=132`) are in effect. You can optionally specify any of the following `COMPARE` command qualifiers as keywords.

DTM CREATE COLLECTION

CHARACTERS	FULL
IGNORE	[NO]PARALLEL
RECORDS	SCREENS
WIDTH	

If you specify more than one keyword, separate the keywords with commas and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses.

The `/NOCOMPARE` qualifier prevents the automatic comparison that DEC/Test Manager ordinarily performs when the collection is executed. Test results for collections created with the `/NOCOMPARE` qualifier can be compared later with the `COMPARE` command.

The default is `/COMPARE`.

/EPILOGUE=file-spec
/NOEPILOGUE

Determines whether the default collection epilogue is run with this collection.

The `/EPILOGUE` qualifier overrides the default collection epilogue file for this collection.

NOTE

The collection epilogue is independent of any test epilogues (epilogues associated with a specified test).

The `/NOEPILOGUE` qualifier cancels any default epilogue for this collection. Use this qualifier to run a collection without a collection epilogue. This qualifier has no effect on individual test epilogues.

By default, the default collection epilogue is run with the collection.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is `/LOG`.

DTM CREATE COLLECTION

/PROLOGUE=file-spec

/NOPROLOGUE

Determines whether the default collection prologue is run with this collection.

The */PROLOGUE* qualifier overrides the default collection prologue file for this collection.

NOTE

The collection prologue is independent of any test prologues (prologues associated with a specified test).

The */NOPROLOGUE* qualifier cancels any default prologue for this collection. Use this qualifier to run a collection without a collection prologue. This qualifier has no effect on individual test prologues.

By default, the default collection prologue is run with the collection.

/SUBMIT[(keyword,...)]

/NOSUBMIT

Determines whether the collection is executed immediately after it is created.

The */SUBMIT* qualifier executes the collection in batch mode immediately after the collection is created. When you enter the */SUBMIT* qualifier, by default the *SUBMIT* command default qualifiers (*/KEEP* and */LOG*) are in effect. You can optionally specify any of the following *SUBMIT* command qualifiers as keywords:

AFTER	[NO]CHARACTERISTICS	CPUTIME
[NO]HOLD	[NO]KEEP	[NO]LOG_FILE
NAME	[NO]NOTIFY	[NO]PRINTER
PRIORITY	QUEUE	[NO]USER
WSDEFAULT	WSEXTENT	WSQUOTA

If you specify more than one keyword, separate the keywords with commas and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses.

The */NOSUBMIT* qualifier creates the collection without submitting it to the batch queue. To run the collection, use the DEC/Test Manager *SUBMIT* command.

The default is */NOSUBMIT*.

DTM CREATE COLLECTION

/TEMPLATE_DIRECTORY=dir-spec
/NOTEMPLATE_DIRECTORY

Determines whether for this collection DEC/Test Manager should search the default template directory for template files. If for a test within the collection a directory is included in the template file specification, DEC/Test Manager always searches that directory for the template file. If a directory is not included, DEC/Test Manager searches the default template directory for the template file.

The */TEMPLATE_DIRECTORY* qualifier overrides for this collection the default template directory and DEC/Test Manager searches the specified directory. The directory you specify can be either another directory or a DEC/CMS library.

The */NOTEMPLATE_DIRECTORY* qualifier overrides for this collection the default template directory. DEC/Test Manager searches your default directory for all template files without directory specifications.

By default, DEC/Test Manager searches the default template directory.

/VARIABLE=(variable-name=variable-value,...)

Overrides the values of the specified global variables for this collection. Valid variable-names are the names for the global variables to be overridden. To override a variable for a collection, the variable must already exist in the DEC/Test Manager library.

An override variable-value is required for each variable appearing on the command line. If you override the value for more than one variable, separate the variables with commas and enclose the list in parentheses. If you override only one variable, you can omit the parentheses.

By default, the collection uses the default values for global variables.

/VERIFY
/NOVERIFY

Specifies whether DEC/Test Manager is to verify the existence of files associated with all test descriptions before creating the collection.

The */VERIFY* qualifier causes DEC/Test Manager to verify that all files associated with all test descriptions in the collection exist. If a referenced file does not exist, DEC/Test Manager does not create the collection.

The */NOVERIFY* qualifier causes DEC/Test Manager to create the collection without verifying file existence. If when the collection executes a file associated with a test description is missing, DEC/Test Manager may not run that test.

DTM CREATE COLLECTION

By default, DEC/Test Manager verifies that all files associated with test descriptions in the collection exist before creating the collection.

Parameter Qualifiers

/GROUP

Identifies the immediately preceding item in the test-group-expression as a group. If a test-group-expression comprises a list, use this qualifier after each item in the list that designates a group.

The default is /TEST_DESCRIPTION.

/TEST_DESCRIPTION

Identifies the immediately preceding item in the test-group-expression as a test expression.

The default is /TEST_DESCRIPTION.

Command Parameters

collection-name

Identifies a set of tests that are run in the same batch job. A collection name can be from 1 to 39 characters long and follows the syntax rules for file names. You cannot use wildcards to specify the collection-name parameter.

The collection-name cannot begin with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

test-group-expression

Specifies items of a test-expression or a group-expression. These include test-names, group-names, and wildcard forms of these names. Items in a test-group-expression are separated by commas.

Identify each item as either a test description or a group with the /GROUP or /TEST_DESCRIPTION parameter qualifiers.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager creates the collection even if you do not provide a remark.

Description

The CREATE COLLECTION command creates a set of files that are used to run the tests in the collection. A collection can contain noninteractive tests, interactive tests, or a combination of interactive and noninteractive tests. You can execute a collection of tests either interactively or in batch.

The CREATE COLLECTION command constructs a set of tests by taking a "snapshot" of the test descriptions at the time the collection is created. Therefore, any changes subsequently made to test descriptions contained in the collection are not reflected in the collection. However, changes made to files referenced by the collection may affect the collection at run time.

NOTE

Collections created with this version of DEC/Test Manager are incompatible with earlier versions of DEC/Test Manager.

You can execute a collection of tests in batch by specifying the /SUBMIT qualifier on the CREATE COLLECTION command or by using the DTM SUBMIT command. If you choose to execute your tests in batch mode, DEC/Test Manager uses the collection-name you specified as the name of the batch job.

You can execute a collection interactively by using the DTM RUN command.

Reexecute a collection with either the RUN or SUBMIT command. Enter the RECREATE command to create the collection again. This incorporates any changes made to the test description since the collection was originally created.

The /BENCHMARK_DIRECTORY and /TEMPLATE_DIRECTORY qualifiers establish a directory other than the defaults established with the SET BENCHMARK_DIRECTORY and SET TEMPLATE_DIRECTORY commands for the benchmark and template files for this collection. You can specify either another directory or a DEC/CMS library.

If a test description you include in a collection does not exist, the collection is not created. The collection is also not created if a variable you specify on the CREATE COLLECTION command line is either not global or does not exist. If you specify the /NOVERIFY qualifier, DEC/Test Manager creates the collection without verifying the existence of files associated with tests in the collection. When the collection is executed, DEC/Test Manager may not execute the tests with missing files. By default, DEC/Test Manager verifies the existence of files associated with the

DTM CREATE COLLECTION

test descriptions in the collection when the collection is created. If a file is missing, the collection is not created.

If on your CREATE COLLECTION command you specify benchmark or template directories different from the library default directories specified with the SET BENCHMARK_DIRECTORY or SET TEMPLATE_DIRECTORY commands, the directories specified on the CREATE COLLECTION command override the directories established with the SET BENCHMARK_DIRECTORY or SET TEMPLATE_DIRECTORY command. However, if you specified different directories for the benchmark or template files when you created a test description, when that test executes the directories associated with the test description are searched for the benchmark and template files.

Examples

1.

```
⌘ DTM CREATE COLLECTION/NOPROLOGUE MARGIN_COLLECT_1 LMTEST*,RMTEST*-  
_⌘"Tests of margin manipulations"  
%DTM-S-CREATED, collection MARGIN_COLLECT_1 created
```

This example creates the collection MARGIN_COLLECT_1. It uses a qualifier to specify that there is no collection prologue, and uses wildcards to specify which tests go into the collection.

2.

```
⌘ DTM CREATE COLLECTION MARGIN_COLLECT_2 LMTEST*, RIGHTMARGIN/GROUP-  
_⌘"More margin tests"  
%DTM-S-CREATED, collection MARGIN_COLLECT_2 created
```

This example creates the collection MARGIN_COLLECT_2. The test-group-expression specifies all tests that begin with LMTEST, and all tests in the group RIGHTMARGIN at the time the collection is created.

DTM CREATE GROUP

Creates an empty group in the DEC/Test Manager library.

Format **DTM CREATE GROUP** *group-name* [*remark*]

Command Qualifier	Default
/[NO]LOG	/LOG

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

group-name

Identifies a group—a category you create to organize tests. A group name can consist of 1 to 39 characters and follows the same syntax rules as for file names. You cannot use wildcards to specify the *group-name* parameter. The *group-name* cannot begin with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

remark

Specifies a string that contains a comment. This parameter is optional. If a *remark* includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a *remark*, you are prompted for one, but DEC/Test Manager creates the group even if you do not provide a *remark*.

DTM CREATE GROUP

Description

The CREATE GROUP command creates an empty group in the DEC/Test Manager library. A group name must be unique among group names in this DEC/Test Manager library. DEC/Test Manager informs you of any error in naming.

After a group is created, you can include test descriptions and other groups in the group with the INSERT TEST_DESCRIPTION and the INSERT GROUP commands.

Example

```
Ⓕ DTM CREATE GROUP LEFTMARGIN "Tests of commands affecting left margin"  
%DTM-S-CREATED, group LEFTMARGIN created
```

This example creates the group LEFTMARGIN and stores a remark.

DTM CREATE LIBRARY

Creates a DEC/Test Manager library in an empty VAX/VMS directory.

Format **DTM CREATE LIBRARY** *directory-spec [remark]*

Command Qualifier	Default
<i>/[NO]LOG</i>	<i>/LOG</i>

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

directory-spec

Specifies an empty directory that you have created with the DCL CREATE/DIRECTORY command. The directory specification must follow VAX/VMS specifications for directory names. Do not specify your current default directory or a directory that contains files.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager creates the library even if you do not provide a remark.

DTM CREATE LIBRARY

Description

The CREATE LIBRARY command creates a DEC/Test Manager library in an empty VAX/VMS directory. The library contains the files that DEC/Test Manager needs to describe, run, and review tests. You have the option of storing your DEC/Test Manager benchmark files and interactive template files in the DEC/CMS library. All other files must be stored outside the DEC/Test Manager library in another directory, or in a DEC/CMS library. If you do this, you must inform DEC/Test Manager where the files are stored.

NOTE

Do not create subdirectories of the directory containing the DEC/Test Manager library. DEC/Test Manager will recognize that they are not part of the library and may delete them. Do not create or modify files in the DEC/Test Manager library and do not delete files from the DEC/Test Manager library. Do not access the DEC/Test Manager library with commands other than DEC/Test Manager commands. Use only DEC/Test Manager Review subsystem commands to access test run output files.

You may find it useful to create a separate library for each project for which you use DEC/Test Manager.

Example

```
Ⓢ CREATE/DIR [PROJECT.DTMLIB]
Ⓢ DTM CREATE LIBRARY [PROJECT.DTMLIB]
%DTM-S-CREATED, DEC/Test Manager library DISKⓈUSER01:[PROJECT.DTMLIB] created
```

This example shows how to create a DEC/Test Manager library by first creating an empty directory, DTMLIB.DIR, and then using the CREATE LIBRARY command to turn this directory into a DEC/Test Manager library.

DTM CREATE TEST_DESCRIPTION

Creates a test description in the DEC/Test Manager library.

Format **DTM CREATE TEST_DESCRIPTION** *test-name [remark]*

Command Qualifiers

/[NO]AUTO_COMPARE
 /BENCHMARK=file-spec
 /COMPARISON_TYPE=keyword
 /EPILOGUE=file-spec
 /FILTER=(keyword,...)
 /INPUT=file-spec
 /[NO]INTERACTIVE
 /[NO]LOG
 /PROLOGUE=file-spec
 /RECORD
 /TEMPLATE=file-spec
 /TERMINATION_CHARACTER=character
 /VARIABLE=(variable-name[=variable-value],...)

Defaults

/AUTO_COMPARE
 /BENCHMARK=test-name.BMK
 None.
 None.
 None.
 None.
 /NOINTERACTIVE
 /LOG
 None.
 None.
 See text.
 /TERMINATION_CHARACTER=CTRL/P
 None.

Command Qualifiers

/AUTO_COMPARE
/NOAUTO_COMPARE

Specifies whether screens are automatically compared when you begin an interactive terminal session. If you specify the */AUTO_COMPARE* qualifier when you have not also specified the */RECORD* or */INPUT* qualifier, DEC/Test Manager ignores */AUTO_COMPARE*.

The default is */AUTO_COMPARE*.

/BENCHMARK=file-spec

Specifies the file to contain the expected output from the test's execution. If this file specification includes a directory specification, it overrides the default benchmark directory for the library. Benchmark files may be located in the DEC/Test Manager library, in another directory, or in a DEC/CMS library.

DTM CREATE TEST_DESCRIPTION

By default, DEC/Test Manager supplies a file-name of the form test-name.BMK.

/COMPARISON_TYPE=keyword

Specifies how the result and benchmark files are to be compared. The valid values for keyword are as follows:

Keyword	Meaning
SCREENS	Compares the files on a screen by screen basis.
CHARACTERS	Compares the screens on a character by character basis.
RECORDS	Compares the screens on a record by record basis.

The SCREENS comparison type has meaning only for interactive tests. If you specify the /COMPARISON_TYPE qualifier to be SCREENS for a noninteractive test, this value is ignored.

By default, a comparison type is not associated with the test description.

/EPILOGUE=file-spec

Adds the specified epilogue file to the test description. The test epilogue file is run immediately after the test template file is executed. This epilogue file is unrelated to the collection epilogue file.

Epilogue files cannot be stored in the DEC/Test Manager library. They should be stored in another directory, or in a DEC/CMS library.

By default, no epilogue file is associated with the test description.

/FILTER=(keyword,...)

Selects one or more filters to remove run-time variables from the result file the collection run produces. The valid values for keyword are as follows.

DTM CREATE TEST_DESCRIPTION

Keyword	Filter
ALL	Specifies that all filters be used
DATE	Replaces date stamps: of the form 1-OCT-86 with dd-mmm-yyyy of the form 11-OCT-86 with dd-mmm-yyyy of the form 9/23/86 with mm/dd/yy of the form September 10, 1986 with month day, year
TIME	Replaces time stamps with hh:mm:ss.xxxx
FILE_NAMES	Replaces the file names with FILENAME.EXT
DIRECTORIES	Replaces the directory specification field in the file specification with DISK:[DIRECTORY]
TRACE_BACK	Replaces 8-bit memory addresses with xxxxxxxx
VERSION	Replaces file versions with VERSION

If you specify more than one keyword, separate the keywords with commas and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses.

By default, no filters are associated with the test description.

/INPUT=file-spec

Creates a SESSION file from an INPUT file. You will see the SESSION file created as DEC/Test Manager records the terminal session while taking input from the INPUT file. The file-spec is required. If you include a file-name but omit the file-type, the file-type defaults to INP.

If you specify both the /RECORD and /INPUT qualifiers, a recording session begins with input being read from the INPUT file until it is exhausted. When the input file is exhausted, the terminal bell rings and the terminal is left in record mode. Otherwise, the recording session terminates when the INPUT file is exhausted.

You cannot specify the /INPUT and /RECORD qualifiers on a CREATE TEST_DESCRIPTION command issued from batch. When DEC/Test Manager records a SESSION file, the terminal characteristics of the recording terminal are associated with the SESSION file. In batch, the terminal characteristics are not available and the command will fail. By default, the SESSION file is recorded in test-name.SESSION. Use the /TEMPLATE qualifier to specify a different name for the SESSION file.

DTM CREATE TEST_DESCRIPTION

/INTERACTIVE ***/NOINTERACTIVE***

Specifies whether the test being created is marked as interactive.

The */INTERACTIVE* qualifier marks a test description as containing an interactive test. Use this qualifier to mark a test description for an interactive test not created with the **CREATE** or **MODIFY TEST_DESCRIPTION/RECORD** command. If you specify the */RECORD* or */INPUT* qualifier, the */INTERACTIVE* qualifier is implied. If you specify the */NOINTERACTIVE* qualifier with either the */RECORD* or */INPUT* qualifier, the */NOINTERACTIVE* qualifier is ignored.

The */NOINTERACTIVE* qualifier marks a test description as containing a noninteractive test.

The default is */NOINTERACTIVE*.

/LOG ***/NOLOG***

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/PROLOGUE=file-spec

Adds the specified prologue file to the test description. The test prologue file is run immediately before the test template file is executed. This prologue is unrelated to the collection prologue.

Prologue files cannot be stored in the DEC/Test Manager library. They should be stored in another directory, or in a DEC/CMS library.

By default, no prologue is associated with the test description.

/RECORD

Begins recording of an interactive terminal session in the specified test template file and creates a test description for it. If you specify a template file with the */TEMPLATE* qualifier, the terminal session is recorded in this file. If you do not include directory specifications for the */TEMPLATE* and */BENCHMARK* qualifiers, the template and benchmark files will be placed in the library default directories when they are created. Otherwise the template and benchmark files will be placed in your default directory. If you specify the */RECORD* qualifier, the */INTERACTIVE* qualifier is implied. If you specify both the */RECORD* and */NOINTERACTIVE* qualifiers, the */NOINTERACTIVE* qualifier is ignored.

DTM CREATE TEST_DESCRIPTION

If you specify both the `/RECORD` and `/INPUT` qualifiers, the recording session first takes input from the specified `INPUT` file. When the `INPUT` file is exhausted, the terminal bell rings and DEC/Test Manager will begin taking input from the terminal. You cannot specify the `/INPUT` and `/RECORD` qualifiers on a `CREATE TEST_DESCRIPTION` command issued from batch. When DEC/Test Manager records a `SESSION` file, the terminal characteristics of the recording terminal are associated with the `SESSION` file. In batch the terminal characteristics are not available and the command will fail.

By default, the interactive terminal session is recorded in the file `test-name.SESSION`. Use the `/TEMPLATE` qualifier to specify a different name for the `SESSION` file.

/TEMPLATE=file-spec

Specifies the command file that runs a test, the file that is the test itself, or the file containing an interactive terminal session. If this file specification includes a directory specification, DEC/Test Manager ignores the default template directory. Template files that you create, except `SESSION` files, cannot be stored in the DEC/Test Manager library. They should be stored in another directory, or in a DEC/CMS library.

By default, DEC/Test Manager supplies a template file name of the form `test-name.COM` for noninteractive tests and `test-name.SESSION` for interactive tests.

/TERMINATION_CHARACTER=character

Specifies a character, which when pressed twice, terminates the recording of an interactive terminal session. When pressed once, the termination character temporarily suspends the recording session to introduce a recording function. For example, `CTRL/P E` is the recording function instructing DEC/Test Manager to terminate automatic screen compare and to begin manual screen compare.

The termination character can be any single character, such as an asterisk (`*`), or more likely, a control character, such as `CTRL/P`. To specify a control character, enter a circumflex (`^`) followed by a letter. For example, to enter the termination character `CTRL/D`, enter a circumflex followed by a `D` (`/TERMINATION_CHARACTER=^D`). You can also specify a termination character by its ASCII decimal representation. For example, `16` can be used to specify `CTRL/P`.

You can also terminate the interactive terminal session by entering the `DCL LOGOUT` command. If you do not want an accounting summary, enter the `DCL STOP/IDENTIFICATION=0` command.

DTM CREATE TEST_DESCRIPTION

The default termination character is CTRL/P.

/VARIABLE=(variable-name[=variable-value],...)

Specifies the variables to associate with the test description. Valid variable-names are names for variables already defined in the DEC/Test Manager library. A variable associated with a test description by this qualifier becomes local in scope. You cannot use wildcards to specify the variable-name.

If you specify more than one variable-name, separate the variable-names with commas and enclose the list in parentheses. If you specify only one variable-name, you can omit the parentheses.

If you specify an optional variable-value, the variable takes on that value for this test description only. If you do not specify a variable-value, the variable retains its default value—the value assigned when the variable is created.

By default, no variables are associated with the test description.

Command Parameters

test-name

Specifies a unique name for a test description. Test-names can be from 1 to 39 characters long and follow the same syntax rules as for file-names. You cannot use wildcards to specify the test-name parameter. The test-name cannot begin with DTM\$ as names with this prefix are reserved for use by DEC/Test Manager.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager creates the test description even if you do not provide a remark.

Description

The CREATE TEST_DESCRIPTION command creates a test description in the DEC/Test Manager library.

The test-name parameter specifies a value for the test-name field of the test description. You can specify values for the other test description fields by supplying values for the appropriate command qualifiers.

DTM CREATE TEST_DESCRIPTION

The test-name is used as the unique identifier of the test description and is the only means of accessing a test description. The /TEMPLATE qualifier specifies the template file for this test. A template file is a command file that is a test or a recorded terminal session. If you do not specify a template file-name at the time you create a test description, DEC/Test Manager supplies a default file-name of the form test-name.COM unless you specified the /INPUT, /INTERACTIVE, or /RECORD qualifiers, in which case the default file-name is test-name.SESSION. The /BENCHMARK qualifier specifies the benchmark file for this test. The benchmark file contains the expected results of the test run. If you do not specify a benchmark file-name at the time you create a test description, DEC/Test Manager supplies a default file-name of the form test-name.BMK. If the file specification for the /BENCHMARK or /TEMPLATE qualifier on a test description specifies a directory other than the collection benchmark or template directory, DEC/Test Manager looks in the specified directory for the template or benchmark file for that test.

The /INPUT qualifier creates a new SESSION file using the specified INPUT file for input. You will see the terminal session take place as DEC/Test Manager records the SESSION file, but you will not be able to enter recording functions at the terminal. If you also specify the /RECORD qualifier, the terminal bell rings and the terminal is left in record mode when the input file is exhausted. Otherwise, the recording session ends when the output file is exhausted. Refer to Appendix B for more information on recording a SESSION file while using INPUT files for input.

When you create a test description, you can include file specifications for the template, benchmark, prologue, or epilogue field regardless of whether the specified file exists. DEC/Test Manager does not check whether the files exist; it simply creates the test description with the file-names you supply or with default file-names.

If you specify a name for the variable field, the variable must exist. If the variable does not exist, DEC/Test Manager will not create the test description.

By default, when you create a collection to run your test, DEC/Test Manager looks for all the files the test description requires. If a file is missing, the collection is not created. If you specify the /NOVERIFY qualifier on the CREATE COLLECTION command, DEC/Test Manager creates the collection without verifying that the files exist.

DTM CREATE TEST_DESCRIPTION

If you use a logical name in a file specification for a CREATE TEST_DESCRIPTION qualifier value, the logical name is not resolved until you include the test in a collection. By default, when the collection is created, DEC/Test Manager resolves the logical name; if DEC/Test Manager cannot find the file, the collection is not created. If you specify the /NOVERIFY qualifier on the CREATE COLLECTION command, DEC/Test Manager creates the collection without resolving the logical name.

Use the COPY TEST_DESCRIPTION command if you need to create several similar test descriptions.

Examples

1.

```
⌘ DTM CREATE TEST_DESCRIPTION/TEMPLATE=LMTEST1.COM LMTEST1-
_⌘"Left margin test"
%DTM-I-DEFAULTED, benchmark file name defaulted to LMTEST.BMK
%DTM-S-CREATED, test description LMTEST1 created.
```

This example creates a test description with the test-name, LMTEST1, and a template file LMTEST1.COM.

2.

```
⌘ DTM CREATE TEST/TEMPLATE=LMTEST2.COM/BENCH=BENCLIB:LMTEST2.BMK
_test name:LMTEST2
_Remark:More left margin tests.
%DTM-S-CREATED, test description LMTEST2 created.
```

This example creates a test description, LMTEST2, and includes a benchmark and a template file-name. Note that DEC/Test Manager prompts for the test-name and remark.

DTM CREATE VARIABLE

Defines a variable in the DEC/Test Manager library.

Format	DTM CREATE VARIABLE	<i>variable-name variable-value [remark]</i>
	Command Qualifiers	Defaults
	/GLOBAL	/LOCAL
	/LOCAL	/LOCAL
	/[NO]LOG	/LOG
	/LOGICAL	/SYMBOL
	/NUMERIC	See text.
	/STRING	See text.
	/SYMBOL	/SYMBOL

Command Qualifiers

/GLOBAL

Defines the variable's scope as accessible to all tests in all collections. You cannot specify both /LOCAL and /GLOBAL on the same CREATE VARIABLE command.

The default is /LOCAL.

/LOCAL

Defines the variable's scope as accessible only to an individual test that references it in its test description. You cannot specify both /LOCAL and /GLOBAL on the same CREATE VARIABLE command.

The default is /LOCAL.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

DTM CREATE VARIABLE

/LOGICAL

Defines the variable's usage as a VAX/VMS logical name. You can abbreviate the */LOGICAL* qualifier only to */LOGI*. Otherwise, it can be confused with the */LOG* qualifier. You cannot specify both */LOGICAL* and */SYMBOL* on the same *CREATE VARIABLE* command.

The default is */SYMBOL*.

/NUMERIC

Defines a symbol's type as a numeric value. Use this qualifier to define a quoted variable-value as numeric. You cannot specify both */NUMERIC* and */STRING* on the same *CREATE VARIABLE* command.

By default, if the variable-value is not enclosed in double quotes ("), the symbol's type is defined as a numeric value.

/STRING

Defines a symbol's type as a string. Use this qualifier to define an unquoted variable-value as a string. You cannot specify both */NUMERIC* and */STRING* on the same *CREATE VARIABLE* command.

By default, if the variable-value is enclosed in double quotes ("), the symbol's type is defined as a string.

/SYMBOL

Defines the variable's usage as a VAX/VMS symbol. When you specify a variable as a symbol with the */SYMBOL* qualifier, you must further define it as either numeric (with the */NUMERIC* qualifier) or string (with the */STRING* qualifier). You cannot specify both */LOGICAL* and */SYMBOL* on the same *CREATE VARIABLE* command.

The default is */SYMBOL*.

Command Parameters

variable-name

Specifies a unique name for a variable. A variable-name can be from 1 to 39 characters long and follows VAX/VMS rules for file-names. You cannot use wildcards to specify the variable-name parameter. You cannot use the variable-names P1 through P8, nor can you use variable-names beginning with DTM\$ as these names are reserved for use by DEC/Test Manager.

variable-value

Specifies the variable's value. The variable has the specified value by default unless its value is redefined for a particular test description.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager creates the variable even if you do not provide a remark.

Description

The CREATE VARIABLE command defines the specified variable in the current DEC/Test Manager library. Only variables defined in the library can be included in test descriptions or collections.

You specify the default value for a variable with the variable-value parameter. If you do not supply a variable-value, DEC/Test Manager prompts you for one. Optional qualifiers allow you to specify how the variable is to be handled during processing. You can override this value for a particular test description.

When defining a variable, you must specify the following information:

- The variable's value
- Whether the variable is local or global
- Whether the variable is a symbol or a logical name
- Whether the variable-value is a string or a numeric value (if the variable is a symbol)

The variable's value is specified with the variable-value parameter. Its use as a local or global variable is specified with the /GLOBAL or /LOCAL qualifier. Its use as a symbol or logical is specified with the /SYMBOL or /LOGICAL qualifier.

If you do not specify the /SYMBOL or /LOGICAL qualifier, DEC/Test Manager creates the variable as a symbol by default. If a variable is defined as a symbol, you must further define the variable-value as either a string by using /STRING, or numeric by using /NUMERIC. If you do not use one of these qualifiers, DEC/Test Manager interprets a quoted variable-value as a string and an unquoted variable-value as numeric. Chapter 6 provides examples and discusses this topic further.

DTM CREATE VARIABLE

Example

```
$ DTM CREATE VARIABLE/SYMBOL/LOCAL INPUT_FILE "emptyfil"-  
_ $"Name of input file, with an empty file as the default"  
%DTM-S-CREATED, symbol variable INPUT_FILE created.
```

This example creates the variable INPUT_FILE. It is defined as a local and as a symbol, and its default value is "emptyfil".

DTM DEFINE/KEY

Defines a keypad key to execute a command-string.

Format **DTM DEFINE/KEY** *key-name "command-string"*

Command Qualifiers	Defaults
/[NO]ECHO	/ECHO
/[NO]IF_STATE=(state-name,...)	Current state.
/[NO]LOCK_STATE	/NOLOCK_STATE
/[NO]SET_STATE=state-name	Current state.
/[NO]TERMINATE	/NOTERMINATE

Command Qualifiers

/ECHO
/NOECHO

Specifies whether the command is echoed after you press the defined key. You cannot define a key by specifying both */NOECHO* and */NOTERMINATE*.

The default is */ECHO*.

/IF_STATE=(state-name,...)
/NOIF_STATE

Specifies a list of states, any one of which must be set to enable the specified key definition. DEC/Test Manager defines the two state-names as *DTM* and *GOLD_DTM*.

The */NOIF_STATE* qualifier selects the current state.

The default is the current state.

/LOCK_STATE
/NOLOCK_STATE

Retains the state specified with the */SET_STATE* qualifier until you use the */SET_STATE* qualifier again to change it.

The default is */NOLOCK_STATE*.

DTM DEFINE/KEY

/SET_STATE=state-name

/NOSET_STATE

Associates a state with the key you are defining. A state-name can be any alphanumeric string. DEC/Test Manager defines the two state-names as DTM and GOLD_DTM. You cannot define a key specifying both */SET_STATE* and */TERMINATE*.

The */NOSET_STATE* qualifier selects the current state.

The default is the current state.

/TERMINATE

/NOTERMINATE

Determines whether the specified command string executes when you press the defined key. When you use the */NOTERMINATE* qualifier, you must press RETURN to execute the command. You cannot define a key specifying both */SET_STATE* and */TERMINATE* or */NOECHO* and */NOTERMINATE*. The default is */NOTERMINATE*.

Command Parameters

key-name

Specifies the key to define. Use the following key-names when defining keys.

Key-name	LK201	VT100	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	,	,	—
MINUS	-	-	—
ENTER	ENTER	ENTER	ENTER
LEFT	←	←	←
RIGHT	→	→	→

Key-name	LK201	VT100	VT52
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	—	—

command-string

Specifies the command-string to be entered when you press the defined key. The command-string can be a DEC/Test Manager command. If the command contains any spaces, enclose it in double quotation marks ("").

Description

The DEFINE/KEY command defines a key on the keypad to execute a DEC/Test Manager command. This command is useful only when you are using DEC/Test Manager in subsystem mode. When you press a keypad key, with one keystroke you enter the command associated with that key. The key definition remains in effect throughout your current DEC/Test Manager session or until you redefine the key. Include your key definitions in the DEC/Test Manager initialization file if you want the key definitions to be active every time you use DEC/Test Manager in subsystem mode.

DEC/Test Manager provides you with a set of default keypad definitions. You can use the DEFINE/KEY command to replace these definitions or to define the undefined keypad keys. Pressing the PF2 key displays the default key definitions.

The state-name value used with the /IF_STATE, /LOCK_STATE, and /SET_STATE qualifiers can be any alphanumeric string. The state-names defined by DEC/Test Manager are DTM and GOLD_DTM. Thus, a single definable key can have two key definitions, depending on the state you use.

DTM DEFINE/KEY

Example

```
DTM> DEFINE/KEY KP4 /IF_STATE=GOLD_DTM /TERMINATE "SET LIBRARY DRA$:[PROJECT.MYLIB]"  
DTM>
```

This command defines the gold function of the 4 key on the DEC/Test Manager keypad to establish the specified DEC/Test Manager library. When you press the GOLD key followed by the keypad 4 key, DEC/Test Manager establishes the specified library.

DTM DELETE COLLECTION

Deletes the specified collection and any associated difference and result files from the DEC/Test Manager library.

Format **DTM DELETE COLLECTION** *collection-expression [remark]*

Command Qualifiers

/[NO]CONFIRM
/[NO]LOG

Defaults

/CONFIRM
/LOG

Command Qualifiers

/CONFIRM
/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each deletion.

The default is /CONFIRM.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

Command Parameters

collection-expression

Specifies the collections to delete. Collection-expression can be a collection-name, a wildcard character, a wildcard in combination with a collection-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted

DTM DELETE COLLECTION

for one, but DEC/Test Manager deletes the collection even if you do not provide a remark.

Description

The DELETE COLLECTION command deletes one or more specified collections and any related difference or result files from the DEC/Test Manager library.

If you are unable to delete a collection and the collection run ended abnormally, use the VERIFY/RECOVER command and then reissue the DELETE COLLECTION command. Also verify that you have sufficient privileges to delete the files. You cannot delete a collection that is in use.

This command does not affect benchmark files.

Examples

1.

```
§ DTM DELETE COLLECTION MARGIN_COLLECT_1 "No longer needed"
Confirm deletion of collection MARGIN_COLLECT_1 [Y/N] (N): y
%DTM-I-DELETED, collection MARGIN_COLLECT_1 deleted
%DTM-S-DELETIONS, 1 deletion completed
```

This example deletes the collection MARGIN_COLLECT_1. The default /CONFIRM qualifier is in effect.

2.

```
§ DTM DELETE COLLECTION *COLLECT_* "Removing all margin tests"
Confirm deletion of collection MARGIN_COLLECT_2 [Y/N] (N): n
%DTM-I-NODELETE, collection MARGIN_COLLECT_2 not deleted
Confirm deletion of collection MARGIN_COLLECT_3 [Y/N] (N): y
%DTM-I-DELETED, collection MARGIN_COLLECT_3 deleted
%DTM-S-DELETIONS, 1 deletion completed
```

This example deletes all the collections that contain COLLECT_ as part of their name. The default /CONFIRM qualifier is in effect. In the first request for confirmation, you reply by pressing RETURN. DEC/Test Manager interprets this as a NO and does not delete the collection. When you confirm the deletion of the second collection by typing Y, DEC/Test Manager deletes this collection.

DTM DELETE GROUP

Deletes a group from the DEC/Test Manager library.

Format **DTM DELETE GROUP** *group-expression [remark]*

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG

Command Qualifiers

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each deletion.

The default is */CONFIRM*.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

group-expression

Specifies the groups to delete. Group-expression can be a group-name, a wildcard character, a wildcard in combination with a group-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager deletes the group even if you do not provide a remark.

DTM DELETE GROUP

Description

The DELETE GROUP command deletes an established group from the DEC/Test Manager library. You cannot delete a group if it contains any test descriptions or other groups, or if it is a subgroup of another group. If any test descriptions or groups are still in the group when you issue the DELETE GROUP command, DEC/Test Manager reports that the specified group has not been deleted.

The SHOW GROUP/MEMBER command lists the groups of which this group is a member. The SHOW GROUP/CONTENTS command lists the groups and test descriptions contained in this group. Use the REMOVE GROUP command to remove subgroups of the group or to remove the group from another group. Use the REMOVE TEST_DESCRIPTION command to remove test descriptions from the group.

Examples

1.

```
Ⓢ DTM DELETE GROUP LEFTMARGIN/NOCONFIRM
%DTM-S-DELETED, group LEFTMARGIN deleted
```

This example deletes the group LEFTMARGIN. You are not prompted for confirmation because the /NOCONFIRM qualifier is in effect.

2.

```
Ⓢ DTM DELETE GROUP * "Deleting all groups"
Confirm deletion of group MATH [Y/N] (N): ALL
%DTM-I-DELETED, group MATH deleted
%DTM-I-DELETED, group RELOP deleted
%DTM-I-DELETED, group VARS deleted
%DTM-S-DELETIONS, 3 deletions completed
```

This example deletes all the groups in the library. The default /CONFIRM qualifier is in effect. By typing ALL, you indicate that all groups can be deleted without further requests for confirmation.

DTM DELETE HISTORY

Deletes the specified history information from the history file.

Format **DTM DELETE HISTORY** *[remark]*

Command Qualifiers	Defaults
/BEFORE=time	See text.
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG
/OUTPUT[=file-spec]	/OUTPUT=HISTORY.OUT

Command Qualifiers

/BEFORE=time

Deletes all history information from the history file dated prior to the specified date. The deleted information is replaced by a single entry stating that history information has been deleted from the history file.

The time can be specified as an absolute, delta, or combination time value, or as one of the following keywords: TODAY, TOMORROW, or YESTERDAY. DEC/Test Manager interprets TOMORROW as the time at which you enter the DELETE HISTORY command.

If you omit the /BEFORE qualifier, the default is to remove information prior to the time you enter the command. If you include the /BEFORE qualifier and do not specify a time, the default is TODAY.

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each deletion.

The default is /CONFIRM.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

DTM DELETE HISTORY

The default is /LOG.

/OUTPUT[=-file-spec]

Sends output from the DELETE HISTORY command to the specified file.

By default, the output is written to a file called HISTORY.OUT in your current default directory.

Command Parameter

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager deletes the history information even if you do not provide a remark.

Description

The DELETE HISTORY command removes the specified history information from the history file. The specified history information is removed from the history file and placed in file HISTORY.DMP in your default directory. After you have removed information from the history file, you cannot replace it. The DELETE HISTORY command is logged in the history file in the usual manner. In addition, it is also logged in the file at the point where history information is being deleted in the following form:

```
date time user-name REMARK "PREVIOUS HISTORY DELETED"
```

Example

```
DTM> DELETE HISTORY /BEFORE=08-JAN "Deleting old information"  
Confirm DELETE HISTORY/BEFORE=8-Jan-1986 [Y/N] (N): Y  
%DTM-S-HISTDEL, 150 history records deleted  
DTM>
```

This example deletes all history records in the history file recorded before January 8, 1986. The deleted information is replaced with the following record:

```
* 8-JAN-1986 00:00:00 SMITH REMARK "PREVIOUS HISTORY DELETED"
```

DTM DELETE TEST_DESCRIPTION

Deletes a test description from the DEC/Test Manager library.

Format **DTM DELETE TEST_DESCRIPTION** *test-expression [remark]*

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG

Command Qualifiers

/CONFIRM
/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each deletion.

The default is /CONFIRM.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

Command Parameters

test-expression

Specifies the test descriptions to delete. Test-expression can be a test-name, a wildcard character, a wildcard in combination with a test-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager deletes the test description even if you do not provide a remark.

DTM DELETE TEST_DESCRIPTION

Description

The DELETE TEST_DESCRIPTION command deletes the specified test description from the library. It also deletes the test description's benchmark file if it exists in the DEC/Test Manager library. If the benchmark file is outside the DEC/Test Manager library (in another directory or in a DEC/CMS library), DEC/Test Manager deletes the test description but does not delete the benchmark file.

You cannot delete a test description if it belongs to a group. Use the REMOVE TEST_DESCRIPTION command to remove a test description from a group. Use the SHOW TEST_DESCRIPTION/GROUPS command to display the groups to which the test description belongs.

Result and difference files generated after a test run are not affected by the DELETE TEST_DESCRIPTION command. See the DELETE COLLECTION command description for information about deleting these files.

Do not delete a test description that is part of an existing collection. If you delete a test description that is part of a collection, you may see error messages when you issue other DEC/Test Manager commands. For example, if you review a collection from which you have deleted a test description and its associated benchmark file for a noninteractive test, you will see a message indicating that the result description contains errors and you will be unable to examine the benchmark file for this result description. If the test description and benchmark file you deleted were for an interactive test using screen comparison, you will be unable to examine any of the files associated with the test.

Examples

1.

```
Ⓕ DTM DELETE TEST_DESC/NOCONFIRM LMTEST1 "Deleting LMTEST1"  
%DTM-S-DELETED, test_description LMTEST1 deleted
```

This example deletes the test description LMTEST1. The /NOCONFIRM qualifier is in effect.

DTM DELETE TEST_DESCRIPTION

2.

```
 $ DTM DELETE TEST_DESC/NOCONFIRM *TEST* "Deleting all tests"
 %DTM-I-DELETED, test_description LMTEST2 deleted
 %DTM-I-DELETED, test_description RMTEST1 deleted
 %DTM-I-DELETED, test_description RMTEST2 deleted
 %DTM-I-DELETED, test_description TEST1 deleted
 %DTM-S-DELETIONS, 4 test_descriptions deleted
```

This example deletes all the test descriptions in the library that contain TEST as part of their name. The /NOCONFIRM qualifier is in effect.

DTM DELETE VARIABLE

DTM DELETE VARIABLE

Deletes the specified variables from the DEC/Test Manager library.

Format **DTM DELETE VARIABLE** *variable-expression [remark]*

Command Qualifiers

/[NO]CONFIRM

/[NO]LOG

Defaults

/CONFIRM

/LOG

Command Qualifiers

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each deletion.

The default is */CONFIRM*.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

variable-expression

Specifies a variable-name, a wildcard character, a wildcard in combination with a variable-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager deletes the variable even if you do not provide a remark.

Description

The DELETE VARIABLE command deletes one or more specified variables from the DEC/Test Manager library.

This command does not delete any variable currently associated with a test description. If you attempt such a deletion, DEC/Test Manager issues an error message. Use the SHOW VARIABLE/TEST_DESCRIPTION command to list the test descriptions with which a variable is associated. Use the MODIFY TEST_DESCRIPTION command to dissociate a variable from a test description.

If you attempt to delete several variables and one or more of them are associated with test descriptions, DEC/Test Manager deletes only those variables not associated with a test description.

Variables are described in Chapter 6.

Example

```
⌘ DTM DELETE VARIABLE INPUT_FILE "Deleting variable INPUT_FILE"  
Confirm deletion of variable INPUT_FILE [Y/N] (N): y  
%DTM-S-DELETED, variable INPUT_FILE deleted.
```

This example deletes the variable INPUT_FILE. The default /CONFIRM qualifier is in effect.

DTM DISPLAY

DTM DISPLAY

Displays the benchmark file for the specified interactive test.

Format **DTM DISPLAY** *test-name*

Command Qualifier

/BENCHMARK

Default

/BENCHMARK

Command Qualifier

/BENCHMARK

Specifies that the benchmark file associated with the specified interactive test is to be displayed.

The default is to display the benchmark file.

Command Parameter

test-name

Specifies the name of the test description for an interactive test whose benchmark file is to be displayed. You cannot use wildcards to specify the test-name parameter.

Description

The DISPLAY command displays the benchmark file for the specified interactive test as it was when it was created. For example, if the benchmark file contains dates or times that were recorded when the test was created, those dates or times are displayed rather than the current date or time. Commands in the benchmark file are not executed.

DEC/Test Manager displays the benchmark file screen by screen. You manipulate the file by pressing keypad keys. This keypad and its use are described in Chapter 3.

The DISPLAY/BENCHMARK command searches for the benchmark file in the default benchmark directory unless this default is overridden by a benchmark directory specified on the test description.

Example

```
DTM> DISPLAY/BENCHMARK MEMO_TEST
```

This command displays the banner screen, Screen 0, for interactive display. Follow the directions on the screen to display your benchmark file. For more information about displaying benchmark files for interactive tests, see Chapter 3.

DTM EXIT

DTM EXIT

Terminates a DEC/Test Manager session.

Format

DTM EXIT

Command Qualifiers

None.

Defaults

None.

Description

The EXIT command terminates a DEC/Test Manager session and returns control to DCL-prompt level. You can also press CTRL/Z to terminate a DEC/Test Manager session.

Example

```
DTM> EXIT  
⚡
```

This example terminates a DEC/Test Manager session.

DTM EXTRACT

Extracts an INPUT file from a SESSION file.

Format **DTM EXTRACT** *session-file-spec [input-file-spec]*

Command Qualifiers

/[NO]LOG
/TERMINATION_CHARACTER=character

Defaults

/LOG
/TERMINATION_CHARACTER=CTRL/P

Command Qualifiers

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

/TERMINATION_CHARACTER=character

Specifies the character that DEC/Test Manager interprets as the termination character when the INPUT file is extracted from the SESSION file. DEC/Test Manager uses this information when translating recording functions in the SESSION file to special strings in the INPUT file.

If you used a termination character other than the default termination character CTRL/P when recording the SESSION file, specify that termination character on the EXTRACT command.

The termination character can be any single character, such as an asterisk (*), or more likely, a control character, such as CTRL/P. To specify a control character, enter a circumflex (^) followed by a letter. For example, to enter the termination character CTRL/D, enter a circumflex followed by a D (/TERMINATION_CHARACTER=`D). You can also specify a termination character by its ASCII decimal representation. For example, 16 can be used to specify CTRL/P.

DTM EXTRACT

Command Parameters

session-file-spec

Specifies an existing SESSION file. If you enter a file-name only, DEC/Test Manager supplies the file-type SESSION. If the SESSION file is in a DEC/CMS library, DEC/Test Manager enters a DEC/CMS FETCH command for the SESSION file element and deletes it when done.

input-file-spec

Specifies the file-spec for the INPUT file to be created. If you omit this parameter, DEC/Test Manager uses the SESSION file-name and supplies the file-type INP. If the INPUT file is to be placed in a DEC/CMS library, DEC/Test Manager either enters a DEC/CMS CREATE ELEMENT command or DEC/CMS RESERVE and REPLACE commands to place the file in the specified DEC/CMS library.

Description

The EXTRACT command extracts an INPUT file from a SESSION file without altering the SESSION file. An INPUT file is a text file that can be edited using the text editor of your choice.

All input, commands, and comments—everything but output and timing records—in the SESSION file are translated and written to the INPUT file. Any braces ({ or }) occurring in the SESSION file are doubled in the INPUT file. Nonprinting characters and sequences are translated based on the terminal characteristics stored in the SESSION file, regardless of the terminal you are using to perform the extraction.

You can use an INPUT file to create a new SESSION file by specifying the INPUT file to be read on the /INPUT qualifier on either the CREATE or MODIFY TEST_DESCRIPTION command. The /INPUT qualifier specifies the INPUT file to be read. When you extract an INPUT file from a SESSION file, you may not be able to recreate the SESSION file except by using a terminal of the same type as the terminal used to record the original terminal session. This is especially true if the recording terminal handles eight-bit characters and the extracting terminal handles seven-bit characters.

Refer to Appendix B for information about using the EXTRACT command.

Example

```
DTM> EXTRACT SAMPLE.SESSION SAMPLE.INP  
DTM-S-EXTRACTED, input file SAMPLE.INP created
```

This example creates the INPUT file SAMPLE.INP. You can now edit this file and use it in conjunction with the CREATE and MODIFY TEST_DESCRIPTION/INPUT commands.

DTM HELP

DTM HELP

Displays HELP text for DEC/Test Manager and Review subsystem commands.

Format **DTM HELP** *[topic]*

Command Qualifiers
None.

Defaults
None.

Command Parameter

topic

Specifies a subject related to DEC/Test Manager or a DEC/Test Manager command about which you want information. Topic can be either a subject (such as RELEASE_NOTES) that is discussed in the DEC/Test Manager HELP file, or a DEC/Test Manager command. The command can include qualifiers.

Description

The HELP command displays DEC/Test Manager information on your screen.

The optional topic parameter allows you to get help on specific topics and on all DEC/Test Manager commands. If you do not specify a topic parameter, you get a display of available HELP features and instructions for displaying the text. If you specify a topic, you see information about the topic displayed. If the topic you enter is a DEC/Test Manager command, you see information about that command.

DTM INSERT GROUP

group-expression2

Specifies the groups into which the subgroups specified in *group-expression1* are to be inserted. A *group-expression* can be a group-name, a wildcard character, a wildcard character in combination with a group-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager inserts the groups even if you do not provide a remark.

Description

The INSERT GROUP command places one or more groups—the subgroups—into one or more other groups. Use this command to create a group hierarchy.

DEC/Test Manager does not insert the same group into another group more than once. If the first group is already a subgroup of the second group, DEC/Test Manager informs you that the INSERT operation has already been done.

DEC/Test Manager does not create recursive group structures. If group B is a subgroup of group A, then group A cannot be a subgroup of group B.

Subsequent changes to tests or subgroups inserted with the INSERT GROUP command are reflected in the group.

For more information on groups, see Chapter 3.

Examples

1.

```
Ⓢ DTM INSERT GROUP LEFTMARGIN MARGINS -  
_Remark: "inserting group LEFTMARGIN into group MARGINS"  
%DTM-S-INSERTED, group LEFTMARGIN inserted into group MARGINS
```

This example inserts the LEFTMARGIN group into the MARGINS group.

DTM INSERT GROUP

2.

```
$ DTM INSERT GROUP RIGHTMARGIN MARGINS,ALL -  
_Remark: "Inserting RIGHTMARGIN into ALL and MARGINS"  
%DTM-I-INSERTED, group RIGHTMARGIN inserted into group ALL  
%DTM-I-INSERTED, group RIGHTMARGIN inserted into group MARGINS  
%DTM-S-INSERTIONS, 2 insertions completed
```

This example inserts the RIGHTMARGIN group into both the ALL and MARGINS groups.

DTM INSERT TEST_DESCRIPTION

The default is /TEST_DESCRIPTION.

/TEST_DESCRIPTION

Identifies the immediately preceding item in a parameter as a test.

The default is /TEST_DESCRIPTION. If you do not include a parameter qualifier after an item, DEC/Test Manager assumes that the item specifies a test description.

Command Parameters

test-group-expression

Specifies the test descriptions or groups of test descriptions that are to be inserted. A test-group-expression can be a test-name, a group-name, a wildcard character, a wildcard character in combination with a test-name or a group-name, or a list of these separated by commas.

Identify the individual items in the test-group-expression as tests or groups using the parameter qualifiers /GROUP and /TEST_DESCRIPTION.

group-expression

Specifies the groups into which the test descriptions are to be inserted. A group-expression can be a group-name, a wildcard character, a wildcard character in combination with a group-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager inserts the test description even if you do not provide a remark.

Description

The INSERT TEST_DESCRIPTION command places one or more test descriptions into one or more groups. DEC/Test Manager will not insert a test description into a group if it is already a member of the group.

You can specify test descriptions in a test-group-expression, identifying the items in the expression as test-names or as group-names using the corresponding parameter qualifier. When you specify a test description, it is inserted into the groups specified in the group-expression parameter.

DTM INSERT TEST_DESCRIPTION

When you specify a group-name on the INSERT TEST_DESCRIPTION command, all the tests that currently belong to that group are inserted into the specified group in their current state. However, the group itself does not become a subgroup of the specified group, and the group-name does not become affiliated with the group named in the group-expression parameter. Subsequent changes made to the individual tests or to the contents of the inserted group are not reflected in the group. You should use the INSERT TEST_DESCRIPTION command with a group-name if you want to include the specified subgroup of tests in their current state and if you do not want subsequent changes reflected in the final group. For more information about using groups, see Chapter 6.

Example

```
$ DTM INSERT TEST_DESCRIPTION MARGINS/GROUP PAGING -  
_Remark: "Inserting tests in MARGINS into PAGING"  
%DTM-I-INSERTED, test description LMTEST1 inserted into group PAGING  
%DTM-I-INSERTED, test description LMTEST2 inserted into group PAGING  
%DTM-I-INSERTED, test description RMTEST1 inserted into group PAGING  
%DTM-I-INSERTED, test description RMTEST2 inserted into group PAGING  
%DTM-S-INSERTIONS, 4 insertions completed
```

This example inserts all the test descriptions in the MARGINS group into the PAGING group. However, the MARGINS group is not inserted.

DTM MODIFY GROUP

Changes information associated with existing groups.

Format **DTM MODIFY GROUP** *group-expression [remark]***Command Qualifiers**

/[NO]LOG

/[NO]REMARK="string"

Defaults

/LOG

See text.

Command Qualifiers**/LOG****/NOLOG**

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

/REMARK="string"**/NOREMARK**

Specifies whether to replace the remark associated with the group. Note that this remark is associated with the group you are modifying; it is not the remark logged with the MODIFY GROUP command.

The /REMARK qualifier adds the specified remark string to the group. This string replaces any previous remark string.

The /NOREMARK qualifier deletes the remark string from the group.

If you do not include either the /REMARK qualifier and a valid remark string or the /NOREMARK qualifier, DEC/Test Manager issues an error message stating that you did not make any changes.

DTM MODIFY GROUP

Command Parameters

group-expression

Specifies the groups to be modified. A group-expression can be a group-name, a wildcard character, a wildcard character in combination with a group-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but the group is modified even if you do not provide a remark. This remark is associated with the MODIFY GROUP command and is logged with it in the history file. It is not the modified remark to be associated with the group.

Description

The MODIFY GROUP command replaces the remarks associated with existing groups. You can modify any number of groups with one MODIFY GROUP command. The remark parameter specifies the remark associated with the MODIFY GROUP command and is logged with it in the history file. The /REMARK qualifier specifies the modified remark to be associated with the modified group. Because the remark is the only item you can modify for a group, DEC/Test Manager displays an error message if you do not supply a value for the remark qualifier or specify the /NOREMARK qualifier.

Example

```
Ⓢ DTM MODIFY GROUP LEFTMARGIN /REMARK="All leftmargin tests"  
_Remark:"Replacing remark for group LEFTMARGIN"  
%DTM-S-MODIFIED, group LEFTMARGIN modified
```

This command removes the remark associated with group LEFTMARGIN, replaces it with "All leftmargin tests", and logs the remark "Replacing remark for group LEFTMARGIN" with the MODIFY GROUP command in the history file.

DTM MODIFY TEST_DESCRIPTION

The default is not to replace the current value for the /AUTO_COMPARE qualifier.

/BENCHMARK=file-spec
/NOBENCHMARK

Specifies whether to replace the benchmark file.

The /BENCHMARK qualifier adds the specified benchmark file specification to the test description. This benchmark file supersedes any benchmark currently associated with the test description.

The new file specification cannot contain a directory specification that resolves to the DEC/Test Manager library. However, you can use DTM\$LIB without a file-name. The new file specification can specify either a directory (other than the DEC/Test Manager library) or a DEC/CMS library. The new file specification overrides any benchmark directory specified with an earlier CREATE TEST_DESCRIPTION command.

The new file specification need not specify the same type of directory as the file specification you are overriding. For example, if you have been storing your benchmark files in a DEC/CMS library, you can specify a directory that is not a DEC/CMS library.

If you specify a file-name for a benchmark file that does not exist, DEC/Test Manager treats the test like a new test when a collection that contains this test executes. When a benchmark file is created, it will have the file-name you specify.

The /NOBENCHMARK qualifier removes the benchmark file specification from the test description and replaces it with the default, test-name.BMK.

The default is not to replace the benchmark file.

/COMPARISON_TYPE=keyword
/NOCOMPARISON_TYPE

Specifies how the result and benchmark files are to be compared. The valid values for keyword are as follows:

Keyword	Meaning
SCREENS	Compares the files on a screen by screen basis.
CHARACTERS	Compares the screens on a character by character basis.
RECORDS	Compares the screens on a record by record basis.

DTM MODIFY TEST_DESCRIPTION

The SCREENS comparison type has meaning only for interactive tests. If you specify the /COMPARISON_TYPE qualifier to be SCREENS for a noninteractive test, this value is ignored.

The /NOCOMPARISON_TYPE qualifier dissociates any comparison type from the test description.

The default is not to replace the current comparison type.

/EPILOGUE=file-spec

/NOEPILOGUE

Specifies whether to replace the test epilogue file.

The /EPILOGUE qualifier adds the specified test epilogue file specification to the test description. The test epilogue file is run whenever the test description is run and does not affect any collection epilogue file run with the test.

The /NOEPILOGUE qualifier removes the current test epilogue file specification from the test description.

The default is not to replace the test epilogue file.

/FILTER=(keyword,...)

/NOFILTER=(keyword,...)

Selects one or more filters to remove run-time variables from the result file produced from the collection run. The valid values for keyword are shown in the following table.

Filter-type	Description
ALL	Specifies that all filters be used
DATE	Replaces date stamps: of the form 1-OCT-86 with dd-mmm-yyyy of the form 11-OCT-86 with dd-mmm-yyyy of the form 9/23/86 with mm/dd/yy of the form September 10, 1986 with month day, year
TIME	Replaces time stamps with hh:mm:ss.xxxx
FILE_NAMES	Replaces the file names with FILENAME.EXT
DIRECTORIES	Replaces the directory specification field in the file specification with DISK:[DIRECTORY]
TRACE_BACK	Replaces 8-bit memory addresses with xxxxxxxx
VERSION	Replaces file versions with VERSION

DTM MODIFY TEST_DESCRIPTION

If you specify more than one keyword, separate the keywords with commas and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses. The `/FILTER` qualifier associates the specified filters with the test description.

The `/NOFILTER` qualifier removes the specified filters from the test description.

By default, the filters associated with the test description are not replaced.

/INPUT=file-spec

Creates a `SESSION` file from an `INPUT` file. You will see the `SESSION` file created as DEC/Test Manager records the terminal session while taking input from the `INPUT` file. The file specification is required. If you include a file-name but omit the file-type, the file-type defaults to `INP`.

If you specify both the `/RECORD` and `/INPUT` qualifiers, a recording session begins with input being read from the `INPUT` file until it is exhausted. When the input file is exhausted, the terminal bell rings and the terminal is left in record mode. Otherwise, the recording session terminates when the `INPUT` file is exhausted.

If you specify the `/TEMPLATE` qualifier with a file specification, the `SESSION` file is recorded in the specified file. Otherwise, the `SESSION` file is recorded in the next version of the test's template file.

/INTERACTIVE ***/NOINTERACTIVE***

Specifies whether the test being created is marked as interactive or noninteractive.

The `/INTERACTIVE` qualifier marks a test description as being an interactive test. Use this qualifier to mark a test description as being an interactive test not created with the `CREATE` or `MODIFY TEST_DESCRIPTION/RECORD` command. If you specify the `/RECORD` or `/INPUT` qualifier, the `/INTERACTIVE` qualifier is implied. If you specify the `/NOINTERACTIVE` qualifier and either the `/RECORD` or `/INPUT` qualifier, the `/NOINTERACTIVE` qualifier is ignored.

The `/NOINTERACTIVE` qualifier marks a test description as containing a noninteractive test.

The default is the current test type.

DTM MODIFY TEST_DESCRIPTION

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/PROLOGUE=file-spec

/NOPROLOGUE

Specifies whether to replace the test prologue file.

The */PROLOGUE* qualifier adds the specified test prologue file specification to the test description. The test prologue is run whenever the test description is run and does not affect any collection prologue run with the test.

The */NOPROLOGUE* qualifier removes the current test prologue file specification from the test description.

The default is not to replace the test prologue file.

/RECORD

Records an interactive terminal session in the specified test template file and creates a test description for it. If you specify a template file with the */TEMPLATE* qualifier, the terminal session is recorded in this file. Otherwise the *SESSION* file is recorded in the next version of the test's template file. If you specify the */RECORD* qualifier, the */INTERACTIVE* qualifier is implied. If you specify both the */RECORD* and */NOINTERACTIVE* qualifiers, the */NOINTERACTIVE* qualifier is ignored.

If you specify both the */RECORD* and */INPUT* qualifiers, a recording session begins with input being read from the *INPUT* file until it is exhausted. The terminal is then left in record mode.

By default, the interactive terminal session is recorded in the next version of the file pointed to by the file specification associated with the */TEMPLATE* qualifier. If you do not include directory specifications for the */TEMPLATE* and */BENCHMARK* qualifiers, the template and benchmark files will be placed in the library default directories when they are created.

/REMARK="string"

/NOREMARK

Specifies whether to replace the remark associated with the test description.

DTM MODIFY TEST_DESCRIPTION

The `/REMARK` qualifier associates the specified remark with the test description being modified. This string replaces any previous remark string. Note that the remark parameter specifies the remark associated with this `MODIFY TEST_DESCRIPTION` command and is logged with it in the history file.

The `/NOREMARK` qualifier deletes the remark string from the test description.

The default is not to replace the remark.

`/TEMPLATE=file-spec`

`/NOTEMPLATE`

Adds the indicated template file specification to the test description. This template supersedes the current template. A test description must always have a template file.

The `/NOTEMPLATE` qualifier specifies a template file name of the form `test-name.COM` for a noninteractive test and `test-name.SESSION` for an interactive test.

The default is not to replace the template file.

`/TERMINATION_CHARACTER=character`

Specifies a character which when pressed twice terminates the recording of an interactive terminal session. When pressed once, the termination character temporarily suspends the recording session to introduce a recording function. For example, `CTRL/P E` is the recording function that instructs DEC/Test Manager to terminate automatic screen compare and to begin manual screen compare.

The termination character can be any single character, such as an asterisk (*), or more likely, a control character, such as `CTRL/P`. To specify a control character, enter a circumflex (^) followed by a letter. For example, to enter the termination character `CTRL/D`, enter a circumflex followed by a D (`TERMINATION_CHARACTER=^D`). You can also specify a termination character by its ASCII decimal representation. For example, 16 can be used to specify `CTRL/P`.

You can also terminate the interactive terminal session by entering the `DCL LOGOUT` command. If you do not want an accounting summary, enter the `DCL STOP/IDENTIFICATION=0` command.

The default termination character is `CTRL/P`.

DTM MODIFY TEST_DESCRIPTION

/VARIABLE=(variable-name[=variable-value],...)

/NOVARIABLE=(variable-name,...)

Associates variables with the test description, changes the values for variables associated with the test description, or dissociates variables from the test description. Valid variable-names are names for existing variables. The */VARIABLE* qualifier associates the specified variables with the test description. These variables must have been previously defined with the *CREATE VARIABLE* command before you can add them to a test description. If the variable does not exist in the DEC/Test Manager library, the command does not proceed. If you want to specify only one variable, you can omit the parentheses. Wildcards cannot be used with variable names.

If you specify a value, DEC/Test Manager changes the variable's value and interprets the variable as having the same usage (symbol or logical) and type (numeric or string) specified at the time the variable is created.

The */NOVARIABLE* qualifier dissociates the specified variables from the test description. If you specify more than one variable, separate the variables with commas and enclose the list in parentheses. If you specify one variable, you can omit the parentheses.

The default is not to replace the variables.

Parameter Qualifiers

/GROUP

Identifies the immediately preceding item in the parameter as a group-expression. If a test-group-expression comprises a list, use this qualifier after each item that designates a group.

The default is */TEST_DESCRIPTION*.

/TEST_DESCRIPTION

Identifies the immediately preceding item in the parameter as a test-expression.

The default is */TEST_DESCRIPTION*.

DTM MODIFY TEST_DESCRIPTION

Command Parameters

test-group-expression

Specifies the test descriptions to be modified. A test-group-expression can be a test-name, a group-name, a wildcard character, a wildcard character in combination with a test-name or a group-name, or a list of these separated by commas.

Identify the individual items in the test-group-expression as tests or groups with the /GROUP and /TEST_DESCRIPTION parameter qualifiers.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager modifies the test descriptions even if you do not provide a remark.

Description

The MODIFY TEST_DESCRIPTION command changes field values for the test descriptions specified in the test-group-expression. You can modify all field values except the test-name.

Command qualifiers allow you to specify which fields in the test description are to be changed and what new values they are to be assigned. You can modify any number of fields with one MODIFY TEST_DESCRIPTION command.

The remark parameter is the remark associated with the MODIFY TEST_DESCRIPTION command and logged with the command in the history file. The /REMARK qualifier is the modified remark to be associated with the modified test description.

The /INPUT qualifier creates a new SESSION file using the specified INPUT file for input. You will see the terminal session take place as . DEC/Test Manager records the SESSION file, but you will not be able to enter recording functions at the terminal. If you also specify the /RECORD qualifier, the terminal bell rings and the terminal is left in record mode when the input file is exhausted. Otherwise, the recording session ends when the input file is exhausted. Refer to Appendix B for more information on recording SESSION file while using INPUT files for input.

Examples

1.

```
‡ DTM MODIFY TEST_DESC/BENCHMARK=BENCLIB:RMTEST1.BMK RMTEST1
  _Remark: "Replacing old benchmark file"
  %DTM-S-MODIFIED, test description RMTEST1 modified.
```

This example replaces a benchmark file specification for the test description RMTEST1.

2.

```
‡ DTM MODIFY TEST_DESC/NOBENCHMARK/TEMPLATE=NEWLMTTEST1.COM LMTEST1
  _Remark: "Replacing template file and removing benchmark file"
  %DTM-S-MODIFIED, test description LMTEST1 modified.
```

This example replaces the template file specification for the test description LMTEST1 and removes the existing benchmark file specification.

DTM MODIFY VARIABLE

DTM MODIFY VARIABLE

Modifies variable definitions in the DEC/Test Manager library.

Format **DTM MODIFY VARIABLE** *variable-expression [remark]*

Command Qualifiers

Command Qualifiers
/GLOBAL
/LOCAL
/[NO]LOG
/LOGICAL
/NUMERIC
/[NO]REMARK="string"
/STRING
/SYMBOL
/VALUE=value

Defaults

Defaults
Current scope.
Current scope.
/LOG
Current usage.
Current type.
Current remark.
Current type.
Current usage.
Current value.

Command Qualifiers

/GLOBAL

Changes the scope of the variable to global. You cannot specify both /LOCAL and /GLOBAL on the same MODIFY VARIABLE command.

/LOCAL

Changes the scope of the variable to local. You cannot specify both /LOCAL and /GLOBAL on the same MODIFY VARIABLE command.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

/LOGICAL

Changes the usage of the variable and defines the variable as a VAX/VMS logical name. You cannot specify both /LOGICAL and /SYMBOL on the

DTM MODIFY VARIABLE

same MODIFY VARIABLE command. You can abbreviate the /LOGICAL qualifier only to /LOGI. Otherwise, it could be confused with the /LOG qualifier.

/NUMERIC

Changes the type of a symbol variable to numeric. You cannot specify both /NUMERIC and /STRING on the same MODIFY VARIABLE command.

/REMARK="string"

/NOREMARK

Determines whether to replace or remove the existing remark.

The /REMARK qualifier replaces any existing remark with the specified remark.

The /NOREMARK qualifier removes the remark string from the variable, leaving the remark field value null.

/STRING

Changes the type of a symbol variable to string. You cannot specify both /NUMERIC and /STRING on the same MODIFY VARIABLE command.

/SYMBOL

Changes the usage of the variable and defines the variable as a VAX/VMS symbol. Variables used as symbols must be further defined as either numeric or string variable types. You cannot specify both /LOGICAL and /SYMBOL on the same MODIFY VARIABLE command.

/VALUE=value

Changes the default value of the variable.

A value specified with the MODIFY VARIABLE command replaces any previous value for the variable.

Command Parameters

variable-expression

Specifies the variables to modify. Variable-expression can be a variable-name, a wildcard character, a wildcard in combination with a variable-name, or a list of these separated by commas.

DTM MODIFY VARIABLE

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager modifies the variables even if you do not provide a remark.

Description

The MODIFY VARIABLE command changes information about variables in the DEC/Test Manager library. Command qualifiers allow you to change specific information about the variables. You can modify any amount of information with one MODIFY VARIABLE command.

Both numeric symbols and logical symbols cannot have a null value. These conditions would generate DCL statements with illegal syntax, such as the following:

```
‡ variable_name =  
‡ define variable_name " "
```

When modifying the type or value of a variable, DEC/Test Manager checks for this illegal condition in the variable fields of all test descriptions.

The remark parameter is the remark associated with the MODIFY VARIABLE command and logged with the command in the history file. The /REMARK qualifier is the modified remark to be associated with the modified variable.

Example

```
‡ DTM MODIFY VARIABLE INPUT_FILE/VALUE=INPUT.RNO  
_Remark: "Replacing value of INPUT_FILE with INPUT.RNO"  
%DTM-S-MODIFIED, variable INPUT_FILE modified.
```

This example assigns a new default value to the variable INPUT_FILE.

DTM PLAY

Executes the specified SESSION file.

Format **DTM PLAY** *file-spec***Command Qualifiers**

/[NO]DISPLAY_TERMINAL=terminal
/[NO]LOG
/[NO]RESULT_FILE=file-spec

Defaults

/DISPLAY_TERMINAL=SYS\$OUTPUT
/LOG
/NORERESULT_FILE

Command Qualifiers

/DISPLAY_TERMINAL=terminal
/NODISPLAY_TERMINAL

Specifies the terminal on which the output is to be displayed.

The default is SYS\$OUTPUT.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

/RESULT_FILE=file-spec
/NORERESULT_FILE

Specifies the file that is to receive a copy of the output. If you specify /RESULT_FILE but do not include a file-spec, DEC/Test Manager places the results in a file named file-name.RES.

The default is /NORERESULT_FILE.

Command Parameter

file-spec

Specifies the file containing the interactive terminal session that is to be executed.

DTM PLAY

Description

The PLAY command executes the specified SESSION file. The file is not part of a collection and the results of the execution are not compared. Use qualifiers to select your output file and the terminal on which to display the run.

The SESSION file runs as if it were executing on the type of terminal on which it was recorded. If the display terminal has characteristics different from the recording terminal, the display you see may not appear as you expect. Even so, the result file is correct.

Example

```
DTM> PLAY MEMO_TEST.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
.
.
.
%DTM-S-CONCLUDED, your interactive test session has concluded
DTM>
```

This command displays a message stating that your interactive terminal session is beginning, followed by the specified SESSION file, and a message stating that your interactive terminal session has concluded.

DTM RECREATE

Recreates a collection using the same command line that originally created it.

Format **DTM RECREATE** *collection-name [remark]*

Command Qualifiers	Defaults
/[NO]CONFIRM	See text.
/[NO]LOG	/LOG

Command Qualifiers

/CONFIRM
/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm that you want to recreate a collection.

The default is to prompt you for confirmation.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

Command Parameters

collection-name

Specifies the test collection that is to be recreated.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, you are prompted for one, but DEC/Test Manager recreates the collection even if you do not provide a remark. This remark is logged with the RECREATE command in the history file. It is not associated with the recreated collection.

DTM RECREATE

Description

The RECREATE command recreates a collection using the CREATE COLLECTION command which originally created the collection. The old collection and all files related to it, except the benchmark file, are deleted from the DEC/Test Manager library and a new collection is created with the latest available versions of each required file.

Thus, if you have changed a test, a variable definition, or something else in the library since you originally created the collection, the recreated collection reflects these changes. This command is also useful if you created your original collection with a collection-expression containing wildcard characters and you subsequently created more tests to be included in the collection.

NOTE

This command will not recreate a collection that you have deleted.

Example

```
DTM> RECREATE TIME "Recreating Collection TIME"  
Collection TIME has not been reviewed, confirm recreation: [Y/N] (N) Y  
%DTM-S-DELETED, collection TIME deleted  
%DTM-S-CREATED, collection TIME created  
%DTM-S-RECREATED, collection TIME has been recreated
```

This command recreates collection TIME. Because you have not reviewed the collection, DEC/Test Manager prompts for confirmation that you want to recreate the collection without first reviewing it.

DTM REMARK

Places a remark in the DEC/Test Manager history file.

Format **DTM REMARK** [*remark*]**Command Qualifier**

/[NO]LOG

Default

/LOG

Command Qualifier**/LOG****/NOLOG**

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

Command Parameter***remark***

Specifies a string that contains a comment. If a remark includes any space characters, it must be enclosed in double quotation marks ("). If you do not provide a remark, a null remark is logged in the history file.

Description

The REMARK command adds a remark to the DEC/Test Manager history file. If you allow DEC/Test Manager to prompt you for a remark, the size of the remark you enter is limited to 254 characters. Otherwise, the total command line length must follow VAX/VMS conventions. The remark is recorded in the history file in the following format:

```
date time user-name REMARK "remark"
```

Use this command to enter a comment about an unusual occurrence in the DEC/Test Manager library.

DTM REMARK

Example

```
* DTM REMARK "Correcting problems created by  
system crash while collection was running"  
%DTM-S-REMARK, remark added to history file
```

You might enter a remark such as this into your history file to clarify your reason for entering a VERIFY/RECOVER command to correct problems with the library.

DTM REMOVE GROUP

Removes one or more groups from one or more other groups.

Format	DTM REMOVE GROUP	<i>group-expression1 group-expression2</i> <i>[remark]</i>
---------------	-------------------------	---

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/LOG

Command Qualifiers

/CONFIRM
/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm each removal.

The default is */CONFIRM*.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

group-expression1

Specifies the groups to be removed. A group-expression can be a group-name, a wildcard character, a wildcard in combination with a group-name, or a list of these separated by commas.

group-expression2

Specifies the groups from which the specified groups are to be removed. A group-expression can be a group-name, a wildcard character, a wildcard in combination with a group-name, or a list of these separated by commas.

DTM REMOVE GROUP

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager removes the groups even if you do not provide a remark.

Description

The REMOVE GROUP command removes one or more groups from one or more other groups, thus reversing the action of the INSERT GROUP command. Removing a group does not delete the group from the DEC/Test Manager library. The REMOVE GROUP command lets you change the group hierarchy by canceling a group's connection as a subgroup of another group.

Examples

1.

```
$ DTM REMOVE GROUP LEFTMARGIN MARGINS /NOCONFIRM -
_ $ "Removing LEFTMARGIN from MARGIN"
%DTM-S-REMOVED, group LEFTMARGIN removed from group MARGINS
```

This example removes the LEFTMARGIN group from the MARGINS group.

2.

```
$ DTM REMOVE GROUP RIGHTMARGIN MARGINS,PAGING /CONFIRM -
_ $ "Removing RIGHTMARGIN from MARGINS and PAGING"
Confirm removal of group RIGHTMARGIN from group PAGING [Y/N] (N) Y
%DTM-I-REMOVED, group RIGHTMARGIN removed from group PAGING
Confirm removal of group RIGHTMARGIN from group MARGINS [Y/N] (N) Y
%DTM-I-REMOVED, group RIGHTMARGIN removed from group MARGINS
$DTM-S-REMOVALS, 2 removals completed
```

This example removes the RIGHTMARGIN group from the PAGING group and the MARGINS group and prompts you to confirm each transaction.

DTM REMOVE TEST_DESCRIPTION

Parameter Qualifiers

/GROUP

Identifies the immediately preceding item in the parameter as a group-expression. If a test-group-expression comprises a list, use this qualifier after each item that designates a group.

The default is /TEST_DESCRIPTION.

/TEST_DESCRIPTION

Identifies the immediately preceding item in the parameter as a test-expression.

The default is /TEST_DESCRIPTION.

Command Parameters

test-group-expression

Specifies the test descriptions to be removed. A test-group-expression can be a test-name, a group-name, a wildcard character, a wildcard character in combination with a test-name or a group-name, or a list of these separated by commas.

Identify the individual items in the test-group-expression as tests or groups with the /GROUP and /TEST_DESCRIPTION parameter qualifiers.

group-expression

Specifies the group from which the test descriptions are to be removed. A group-expression can be a group-name, a wildcard, a wildcard in combination with a group-name, or a list of these separated by commas.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager removes the test descriptions even if you do not provide a remark.

DTM REMOVE TEST_DESCRIPTION

Description

The REMOVE TEST_DESCRIPTION command removes one or more test descriptions from one or more groups, thus reversing the action of the INSERT TEST_DESCRIPTION command. It does not delete the test description from the DEC/Test Manager library.

Example

```
⌘ DTM REMOVE TEST_DESCRIPTION MARGINS/GROUP PAGING /NOCONFIRM -  
_⌘ "Removing MARGINS from PAGING"  
%DTM-I-REMOVED, test_description LMTEST1 removed from group PAGING  
%DTM-I-REMOVED, test_description LMTEST2 removed from group PAGING  
%DTM-I-REMOVED, test_description RMTEST1 removed from group PAGING  
%DTM-I-REMOVED, test_description RMTEST2 removed from group PAGING  
%DTM-S-REMOVALS, 4 removals completed
```

This example removes all the tests in the MARGINS group from the PAGING group without prompting for confirmation.

DTM REVIEW

DTM REVIEW

Invokes the Review subsystem to examine and manipulate test results for a collection of tests.

Format **DTM REVIEW** *collection-name [remark]*

Command Qualifier

/[NO]READ_ONLY

Default

/NOREAD_ONLY

Command Qualifiers

/READ_ONLY

/NOREAD_ONLY

Determines whether you are a primary or read-only user of the Review subsystem. There can be only one primary reviewer at a time. This reviewer can enter all Review subsystem commands. There can be multiple read-only reviewers who cannot make changes to the result descriptions or to the files they describe. Read-only reviewers cannot enter the INSERT or UPDATE commands.

The default is */NOREAD_ONLY*.

Prompts

DTM_REVIEW>

Command Parameters

collection-name

Identifies a collection created with the CREATE COLLECTION command.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks ("). The collection is reviewed even if you do not provide a remark.

Description

The REVIEW command invokes the Review subsystem and allows you to examine and manipulate test results for the specified collection. You cannot use wildcards when specifying the collection-name parameter. A collection must be run and compared before it can be reviewed. A summary of the results for each test description is stored in the test's result description. Note that a test's result-description-name is the same as its test-name and that the result descriptions are alphabetically ordered.

If you enter the REVIEW command without the /READ_ONLY qualifier, you are designated as the **primary reviewer** of the collection. Only one person at a time can be the primary reviewer. This reviewer can use all Review subsystem commands.

If you enter the REVIEW/READ_ONLY command, you are designated as a **read-only reviewer** of the collection. You can browse through the result descriptions and print files, but you cannot make any changes to the result descriptions. Read-only reviewers cannot enter the UPDATE or INSERT commands. There can be multiple read-only reviewers for a collection.

If the primary reviewer is updating benchmark files when a read-only reviewer enters a SHOW command, the resulting display may be inaccurate. Under the same circumstances, the Collection Summary Information also may be incorrect and files queued for printing may disappear. A read-only reviewer can lessen the probability that a file marked for printing will disappear by using the PRINT/NOW or PRINT/SELECTED command. The /NOW qualifier causes DEC/Test Manager to queue for printing only the files specified on the current command line. The /SELECTED qualifier causes DEC/Test Manager to queue for printing all files specified during the current review session, including those specified on the current command line.

The following commands are valid within the Review subsystem:

- BACK, FIRST, LAST, NEXT, and SELECT—position you at result descriptions
- PRINT and SHOW—display the test result files referenced by the result descriptions
- INSERT and UPDATE—allow you to manipulate the test result files referenced by the result descriptions
- PCA—allows you to invoke the Analyzer of the VAX Performance and Coverage Analyzer

DTM REVIEW

- ATTACH and SPAWN—allow you to create and move between subprocesses of your current Review session
- HELP, DEFINE/KEY, @file-spec, and EXIT—display HELP information, redefine the keypad keys, execute commands in a specified command file, and exit the Review subsystem

When you enter the Review subsystem, DEC/Test Manager displays the Collection Summary Information followed by the Review subsystem prompt. The Collection Summary Information includes:

- The collection-name
- The number of tests in the collection
- The date and time the collection was created
- The command that created the collection
- Whether the collection was reviewed previously and, if so, when
- The number of tests whose comparisons aborted
- The number of new tests
- The number of tests with successful comparisons
- The number of tests with unsuccessful comparisons
- The number of updated tests
- The number of tests that did not run

You can redisplay the Collection Summary Information at any time while you are in the Review subsystem by issuing the SHOW/SUMMARY command.

When entering the Review subsystem, you are not positioned at a result description. At this point, you can press RETURN to position yourself at the first result description, or you can issue a subsystem positioning command to select a specific result description.

When reviewing the results for a collection of tests, it is helpful to note the following:

- Each result description can be identified both by its result-description-name and by its comparison status.
- By specifying either result-description-expressions containing wild-cards, or by specifying one or more comparison status qualifiers (/COMPARISON_ABORTED, /NEW, /NOT_RUN, /SUCCESSFUL, /UNSUCCESSFUL, and /UPDATED), or by specifying both, you can consider sets of result descriptions with similar characteristics.

- For the SHOW and PRINT commands, you can also specify output file qualifiers (/BENCHMARK, /DIFFERENCE, and /RESULT) to print or display output files. For example, you can enter the SHOW * /SUCCESS /BENCHMARK command to display the benchmark files for all successful tests.
- When you enter a Review subsystem command that takes the result-description-expression and the comparison status qualifiers, the following information applies:
 - If you omit the result-description-expression and the comparison status qualifiers, DEC/Test Manager performs the operation on the current result description.
 - If you specify a result-description-name and omit the comparison status qualifiers, DEC/Test Manager performs the operation on the specified result description and makes it the current result description.
 - If you specify several result descriptions by entering a result-description-expression and you omit the comparison status qualifiers, DEC/Test Manager performs the operation on all result descriptions matching the result-description-expression, but it *does not* change the current result description.
 - If you specify a single result-description-name and one or more comparison status qualifiers, DEC/Test Manager performs the operation on the specified result description, makes it the current result description, and ignores the comparison status qualifiers.
 - If you omit the result-description-expression parameter and include one or more comparison status qualifiers, DEC/Test Manager interprets this as the asterisk (*) wildcard parameter and operates on all result descriptions with the specified comparison status or statuses.
 - If you include both a result-description-expression and one or more comparison status qualifiers, DEC/Test Manager operates on all result descriptions that match both the result-description-expression and one of the comparison status qualifiers.

This general information applies to most Review subsystem commands. See the individual command descriptions for more information.

DTM REVIEW

NOTE

A result-description-expression can consist of a result-description-name, a wildcard character, or one or more wildcard characters used in combination with a full or partial result-description-name. A result-description-expression cannot be a list of these separated by commas.

If you invoked the Collector of the VAX Performance and Coverage Analyzer when you ran your DEC/Test Manager tests, you can issue the PCA command from the Review subsystem to invoke the Analyzer. From the Analyzer, you can examine the performance and coverage data gathered by the Collector. The PCA command is described in this chapter. Chapter 7 describes the use of DEC/Test Manager with the VAX Performance and Coverage Analyzer.

Example

```
$ DTM REVIEW MARGIN_COLLECT_1
Collection MARGIN_COLLECT_1 with 7 tests created was on
04-JAN-1986 15:23:32 by the command:
    CREATE COLLECTION MARGIN_COLLECT_1/NOPROLOGUE LMTEST*, RMTEST*
"Tests of margin manipulations"
  Last Review Date = 04-JAN-1986
  Success count = 3
  Unsuccessful count = 2
  New test count = 1
  Updated test count = 1
  Comparisons aborted = 0
  Test not run count = 0

DTM_REVIEW>
```

This example invokes the Review subsystem for the collection MARGIN_COLLECT_1.

Subsystem Commands

The following is a list of DEC/Test Manager Review subsystem commands.

Subsystem Command Summary

@file-spec
ATTACH process-name
 /IDENTIFICATION
 /PARENT
BACK [count]
 /COMPARISON_ABORTED
 /NEW
 /NOT_RUN
 /SUCCESSFUL
 /UNSUCCESSFUL
 /UPDATED
DEFINE/KEY key-name "command-string"
 /[NO]ECHO
 /[NO]IF_STATE
 /[NO]LOCK_STATE
 /[NO]SET_STATE
 /[NO]TERMINATE
EXIT
 /[NO]INSERT
 /[NO]PRINT
FIRST
HELP topic
INSERT [result-description-expression]
 /COMPARISON_ABORTED
 /[NO]CONFIRM
 /NEW
 /NOT_RUN
 /SUCCESSFUL
 /UNSUCCESSFUL
 /UPDATED
LAST
NEXT [count]
 /COMPARISON_ABORTED
 /NEW
 /NOT_RUN
 /SUCCESSFUL
 /UNSUCCESSFUL
 /UPDATED

DTM REVIEW

Subsystem Command Summary (Cont.)

PCA
PRINT [result-description-expression]
 /BENCHMARK
 /COMPARISON_ABORTED
 /DIFFERENCE
 /[NO]LOG
 /NEW
 /NOT_RUN
 /NOW
 /RESULT
 /SUCCESSFUL
 /UNSUCCESSFUL
 /UPDATED
SELECT result-description-name
SHOW [result-description-expression]
 /BENCHMARK
 /COMPARISON_ABORTED
 /DIFFERENCE
 /FILES
 /NEW
 /NOT_RUN
 /OUTPUT
 /RESULT
 /SUCCESSFUL
 /SUMMARY
 /UNSUCCESSFUL
 /UPDATED
SPAWN [command]
 /[NO]CARRIAGE_CONTROL
 /[NO]CLI
 /INPUT
 /[NO]KEYPAD
 /[NO]LOG
 /[NO]LOGICAL_NAMES
 /[NO]NOTIFY
 /OUTPUT
 /PROCESS
 /[NO]PROMPT
 /[NO]SYMBOLS
 /[NO]WAIT

Subsystem Command Summary (Cont.)

UPDATE [result-description-expression]
/[NO]CONFIRM
/[NO]LOG

DTM_REVIEW > @file-spec

DTM_REVIEW > @file-spec

Executes DEC/Test Manager commands contained in the specified file.

Format **DTM_REVIEW > @file-spec**

Command Qualifiers
None.

Defaults
None.

Command Parameter

file-spec

Specifies the command procedure you want to execute. If the *file-spec* does not include a file-type, DEC/Test Manager assumes the default file-type COM.

Description

The @file-spec command executes the commands in the specified file. The file can contain any DEC/Test Manager command, including another @file-spec command. Do not preface the commands with DTM or \$. For example, enter BACK 5 not DTM BACK 5.

When DEC/Test Manager executes an EXIT command or reaches the end of the command procedure, it leaves you in your current position. The invoking command stream can be either the terminal or another command procedure.

DTM_REVIEW > ATTACH

Switches control from your current process to another process in your job.

Format **DTM_REVIEW > ATTACH** *process-name*

Command Qualifiers	Defaults
/IDENTIFICATION= <i>pid</i>	None.
/PARENT	None.

Command Qualifiers***/IDENTIFICATION=*pid****

Specifies the process identification (PID) of the process to which you want to attach your terminal. You can omit the leading zeros when you specify a PID.

If you specify the */IDENTIFICATION* qualifier, do not specify the process name parameter or the */PARENT* qualifier. If you do not specify a qualifier, you must specify a process name.

/PARENT

Specifies that the process you want to attach to is your original, or parent, process.

If you specify the */PARENT* qualifier, do not specify the process name parameter or the */IDENTIFICATION* qualifier. If you do not specify a qualifier, you must specify a process name.

Command Parameter***process-name***

Specifies an existing process to which you want to attach your terminal.

If you specify either the */IDENTIFICATION* or */PARENT* qualifier, do not specify the process name parameter or the other qualifier. If you do not specify a qualifier, you must specify a process name.

DTM_REVIEW > ATTACH

Description

The ATTACH command allows you to connect your terminal to another process in your job. You can use the ATTACH command to change control between subprocesses you have created with the SPAWN command or to reconnect to your parent process. You can also use the ATTACH command in conjunction with the SPAWN/WAIT command to return to your Review session without terminating the subprocess. See the SPAWN command for more information.

DTM_REVIEW > BACK

Moves you backward through the result descriptions being reviewed.

Format **DTM_REVIEW > BACK** *[count]***Command Qualifiers**

/COMPARISON_ABORTED
/NEW
/NOT_RUN
/SUCCESSFUL
/UNSUCCESSFUL
/UPDATED

Defaults

Previous result description.
Previous result description.
Previous result description.
Previous result description.
Previous result description.
Previous result description.

Command Qualifiers***/COMPARISON_ABORTED***

Moves you to the previous result description for a test whose comparison aborted.

The default is to move to the previous result description.

/NEW

Moves you to the previous result description for a new test.

The default is to move to the previous result description.

/NOT_RUN

Moves you to the previous result description for a test that did not run.

The default is to move to the previous result description.

/SUCCESSFUL

Moves you to the previous successfully compared result description.

The default is to move to the previous result description.

/UNSUCCESSFUL

Moves you to the previous unsuccessfully compared result description.

DTM_REVIEW > BACK

The default is to move to the previous result description.

/UPDATED

Moves you to the previous updated result description.

The default is to move to the previous result description.

Command Parameter

count

An integer that indicates the number of result descriptions to move backward from the current result description. The default is 1.

Description

The BACK command moves you backward through the result descriptions being reviewed. The optional count parameter is the number of result descriptions to move backward from the current result description. If the parameter is omitted, you move back to the previous result description. You cannot use the BACK command to move back beyond the first result description. The result description to which you move backward becomes the current result description.

The comparison status qualifiers */COMPARISON_ABORTED*, */NEW*, */NOT_RUN*, */SUCCESSFUL*, */UNSUCCESSFUL*, and */UPDATED* move you back to the most recent result description with the specified comparison status. Specifying a count parameter in combination with a comparison status qualifier moves you back through the specified number of result descriptions with the specified comparison status.

DTM_REVIEW > DEFINE/KEY

Defines a keypad key to execute a command-string.

Format **DTM_REVIEW > DEFINE/KEY** *key-name "command-string"***Command Qualifiers**

/[NO]ECHO
/[NO]IF_STATE=(state-name,...)
/[NO]LOCK_STATE
/[NO]SET_STATE=state-name
/[NO]TERMINATE

Defaults

/ECHO
Current state.
/NOLOCK_STATE
Current state.
/NOTERMINATE

Command Qualifiers

/ECHO
/NOECHO

Specifies whether the command is echoed after you press the defined key. You cannot define a key by specifying both **/NOECHO** and **/NOTERMINATE**.

The default is **/ECHO**.

/IF_STATE=(state-name,...)
/NOIF_STATE

Specifies a list of states, any one of which must be set to enable the specified key definition. DEC/Test Manager defines the two states **REVIEW** and **GOLD_REVIEW**.

The **/NOIF_STATE** qualifier selects the current state.

The default is the current state.

/LOCK_STATE
/NOLOCK_STATE

Retains the state specified with the **/SET_STATE** qualifier until you use the **/SET_STATE** qualifier again to change it.

The default is **/NOLOCK_STATE**.

DTM_REVIEW > DEFINE/KEY

/SET_STATE=state-name

/NOSET_STATE

Associates a state with the key you are defining. A state-name can be any alphanumeric string. DEC/Test Manager defines the two states REVIEW and GOLD_REVIEW. You cannot define a key specifying both /SET_STATE and /TERMINATE.

The /NOSET_STATE qualifier selects the current state.

The default is the current state.

/TERMINATE

/NOTERMINATE

Determines whether the specified command string executes when you press the defined key. When you use /NOTERMINATE, you must press RETURN to execute the command. You cannot define a key specifying both /SET_STATE and /TERMINATE or /NOECHO and /NOTERMINATE.

The default is /NOTERMINATE.

Command Parameters

key-name

Specifies the key to define. Use the following key-names when defining keys.

Key-name	LK201	VT100	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	,	,	—
MINUS	-	-	—
ENTER	ENTER	ENTER	ENTER
LEFT	←	←	←

DTM_REVIEW > DEFINE/KEY

Key-name	LK201	VT100	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	—	—

command-string

Specifies the command-string to be entered when you press the defined key. The command-string can be a Review subsystem command. If the command contains any spaces, enclose it in double quotation marks ("").

Description

The DEFINE/KEY command defines a key on the keypad to execute a Review subsystem command. When you press a keypad key, with one or two keystrokes you enter the command associated with that key. The key definition remains in effect throughout your current Review session or until you redefine the key. Keys defined during a Review session do not affect the DEC/Test Manager subsystem level key definitions. Include your key definitions in the DEC/Test Manager initialization file if you want the key definitions to be active every time you use the Review subsystem.

DEC/Test Manager provides you with a set of default keypad definitions for the Review subsystem. You can use the DEFINE/KEY command to replace these definitions or to define the undefined keypad keys. Pressing the PF2 key displays the default key definitions.

The state-name parameter used with the /IF_STATE, /LOCK_STATE, or /SET_STATE qualifiers can be any alphanumeric string. The state-names defined by DEC/Test Manager for the Review subsystem are REVIEW and GOLD_REVIEW. Thus, a single definable key can have two key definitions, depending on the state you use.

DTM_REVIEW> DEFINE/KEY

Example

```
DTM_REVIEW> DEFINE/KEY KP1 /IF_STATE=REVIEW /TERMINATE "NEXT/SUCCESSFUL"  
DTM_REVIEW> DEFINE/KEY KP1 /IF_STATE=GOLD_REVIEW /TERMINATE "BACK/SUCCESSFUL"  
DTM_REVIEW>
```

This example defines two keys on the Review subsystem keypad. When you press the keypad 1 key, you move forward to the next result description for a successful test. When you press the GOLD key followed by the keypad 1 key, you move backward to the previous result description for a successful test.

DTM_REVIEW > EXIT

Terminates the Review session and returns control to the existing command level when you invoked the Review subsystem.

Format **DTM_REVIEW > EXIT****Command Qualifiers**

/NOINSERT
/NOPRINT

Defaults

Inserts test descriptions into group.
Prints files.

Command Qualifiers***/NOINSERT***

Specifies that a group is not to be created from tests marked with the INSERT command when you exit the Review subsystem.

/NOPRINT

Specifies that files marked with the PRINT command are not to be printed when you exit the Review subsystem.

Description

The EXIT command terminates the Review session and returns control to the existing command level when you invoked the Review subsystem. Unless you specify */NOINSERT* or */NOPRINT*, tests marked with the INSERT command are placed in a group and files selected with the PRINT command are placed in the system's default printer queue.

You can also press CTRL/Z to leave the Review subsystem as if you had entered the EXIT command. You can press CTRL/C to leave the Review subsystem as if you had entered the EXIT/*/NOINSERT*/*/NOPRINT* command.

DTM_REVIEW > FIRST

DTM_REVIEW > FIRST

Moves you to the first result description in the collection you are reviewing.

Format **DTM_REVIEW > FIRST**

Command Qualifiers

None.

Defaults

None.

Description

The FIRST command moves you to the first result description in the collection you are reviewing. The first result description becomes the current result description.

DTM_REVIEW > HELP

Displays HELP text for Review subsystem commands.

Format **DTM_REVIEW > HELP** *[topic]***Command Qualifiers**

None.

Defaults

None.

Command Parameter***topic***

Specifies a subject about which you want information. A topic can be either a subject (such as KEYPAD) that is discussed in the Review subsystem HELP file, or a Review subsystem command. (The command can include qualifiers.) A list of Review subsystem HELP topics appears after you type HELP.

Description

The HELP command displays Review subsystem information on your screen.

The optional topic parameter allows you to get help on specific topics and on all Review subsystem commands. If you do not specify a topic, you get a display of available HELP features and instructions for displaying the HELP text. If you specify a topic, information is displayed about that topic. If you specify a Review subsystem command, information is displayed about that command.

DTM_REVIEW > INSERT

DTM_REVIEW > INSERT

Marks test descriptions for insertion into a group DEC/Test Manager creates when you exit from the Review subsystem.

Format DTM_REVIEW > INSERT

Command Qualifiers

/COMPARISON_ABORTED
/[NO]CONFIRM
/NEW
/NOT_RUN
/SUCCESSFUL
/UNSUCCESSFUL
/UPDATED

Defaults

Current result description.
/NOCONFIRM
Current result description.
Current result description.
Current result description.
Current result description.
Current result description.

Command Qualifiers

/COMPARISON_ABORTED

Marks for inclusion test descriptions for tests whose comparisons aborted.

The default is to mark the test description associated with the current result description.

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm the processing of each test description marked for insertion.

The default is */NOCONFIRM*.

/NEW

Marks for inclusion test descriptions for new tests.

The default is to mark the test description associated with the current result description.

/NOT_RUN

Marks for inclusion test descriptions for tests that did not run.

The default is to mark the test description associated with the current result description.

/SUCCESSFUL

Marks for inclusion test descriptions for tests that compared successfully.

The default is to mark the test description associated with the current result description.

/UNSUCCESSFUL

Marks for inclusion test descriptions for tests that compared unsuccessfully.

The default is to mark the test description associated with the current result description.

/UPDATED

Marks for inclusion test descriptions for tests whose benchmark files have been updated.

The default is to mark the test description associated with the current result description.

Command Parameter***result-description-expression***

Specifies one or more test descriptions for insertion into a group that is created when you exit from the Review subsystem. The result-description-name for a test is the same as its test name.

Description

The INSERT command marks test descriptions for insertion into a group that DEC/Test Manager creates when you exit from the Review subsystem. All test descriptions selected with the INSERT command during a Review session are inserted into the same group. One group is created for each review session. To create a second group, exit from the Review subsystem, initiate another Review session for the same collection, and create another group. The EXIT/NOINSERT command specifies that the marked test descriptions are not to be inserted into a group when you exit from the Review subsystem. The group is not created.

DTM_REVIEW > INSERT

DEC/Test Manager supplies the group-name based on the name of the collection you are reviewing and the number of times it has been reviewed. The format is

`collection-name$DTM_#`

The pound sign (#) specifies the number of times the collection has been reviewed.

Omitting the result-description-expression parameter marks the current test description for inclusion in the group. Specifying a result-description-name marks for inclusion that test description only. That result description becomes the current result description. Specifying a result-description-expression containing wildcard characters marks for inclusion all tests whose result-description-names match the result-description-expression. The current result description is not changed.

The comparison status qualifiers /COMPARISON_ABORTED, /NEW, /NOT_RUN, /SUCCESSFUL, /UNSUCCESSFUL, and /UPDATED mark for inclusion all result descriptions with the specified comparison status. Including one or more comparison status qualifiers and a result-description-expression marks for inclusion all test descriptions that match both the result-description-expression and one of the qualifiers.

DTM_REVIEW > LAST

Moves you to the last result description in the collection you are reviewing.

Format **DTM_REVIEW > LAST**

Command Qualifiers
None.

Defaults
None.

Description

The LAST command moves you to the last result description of the collection you are reviewing—the current collection. The last result description becomes the current result description.

DTM_REVIEW > NEXT

DTM_REVIEW > NEXT

Moves you forward through the result descriptions being reviewed.
Pressing RETURN also moves you forward to the next result description.

Format DTM_REVIEW > NEXT [count]

Command Qualifiers

/COMPARISON_ABORTED
/NEW
/NOT_RUN
/SUCCESSFUL
/UNSUCCESSFUL
/UPDATED

Defaults

Next result description.
Next result description.
Next result description.
Next result description.
Next result description.
Next result description.

Command Qualifiers

/COMPARISON_ABORTED

Moves you to the next result description for a test whose comparison aborted.

The default is to move to the next result description.

/NEW

Moves you to the next result description for a new test.

The default is to move to the next result description.

/NOT_RUN

Moves you to the next result description for a test that did not run.

The default is to move to the next result description.

/SUCCESSFUL

Moves you forward to the next successfully compared result description.

The default is to move to the next result description.

/UNSUCCESSFUL

Moves you forward to the next unsuccessfully compared result description.

The default is to move to the next result description.

/UPDATED

Moves you forward to the next result description whose benchmark file has been updated.

The default is to move to the next result description.

Command Parameter***count***

An integer that indicates the number of result descriptions to move forward from the current result description. The default is 1.

Description

The NEXT command moves you forward through the result descriptions being reviewed. The optional count parameter is the number of result descriptions to move forward from the current result description. If the parameter is omitted, you move forward to the next result description. You cannot use the NEXT command to move beyond the last result description. The result description to which you move forward becomes the current result description.

The comparison status qualifiers */COMPARISON_ABORTED*, */NEW*, */NOT_RUN*, */SUCCESSFUL*, */UNSUCCESSFUL*, and */UPDATED* move you forward to the next result description with the specified comparison status. Specifying a count parameter in combination with a comparison status qualifier moves you forward through the specified number of result descriptions with the specified comparison status.

Pressing RETURN at the DTM_REVIEW> prompt also moves you to the next result description.

DTM_REVIEW> PCA

DTM_REVIEW> PCA

Invokes the Analyzer of the VAX Performance and Coverage Analyzer.

Format DTM_REVIEW> PCA

Command Qualifiers

None.

Defaults

None.

Description

The PCA command invokes the Analyzer of the VAX Performance and Coverage Analyzer. You must be positioned at a result description before you issue the PCA command.

The PCA command spawns a subprocess that invokes the Analyzer, specifies the default data file set up by DEC/Test Manager as input to the Analyzer, and sets up an Analyzer filter that includes only performance and coverage data collected while the current test was run. The subprocess spawned by the PCA command has all the globally defined symbols of your current process.

If the collection of tests you are reviewing was not run with the Collector, the Analyzer issues an error message when you enter the PCA command. The error message states that the expected data file does not exist. After the error message, you are returned to the Analyzer prompt (PCAA>). See Chapter 7 for information about using DEC/Test Manager with the VAX Performance and Coverage Analyzer. See the *VAX Performance and Coverage Analyzer User's Reference Manual* for information about the VAX Performance and Coverage Analyzer.

DTM_REVIEW> PRINT

Selects one or more files for printing.

Format **DTM_REVIEW> PRINT** *[result-description-expression]***Command Qualifiers**

/BENCHMARK
 /COMPARISON_ABORTED
 /DIFFERENCE
 /[NO]LOG
 /NEW
 /NOT_RUN
 /NOW
 /RESULT
 /SELECTED
 /SUCCESSFUL
 /UNSUCCESSFUL
 /UPDATED

Defaults

/RESULT
 Current result description.
 /RESULT
 /LOG
 Current result description.
 Current result description.
 See text.
 /RESULT
 See text.
 Current result description.
 Current result description.
 Current result description.

Command Qualifiers***/BENCHMARK***

Prints benchmark files from the specified result descriptions.

The default is /RESULT.

/COMPARISON_ABORTED

Prints files from result descriptions for tests whose comparisons aborted.

The default is to select files from the test description associated with the current result description.

/DIFFERENCE

Prints difference files from the specified result descriptions.

The default is /RESULT.

DTM_REVIEW> PRINT

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/NEW

Prints files from result descriptions for new tests.

The default is to select files from the test description associated with the current result description.

/NOT_RUN

Prints files from result descriptions for tests that did not run. When a test does not run, its result description may or may not contain a benchmark file.

The default is to select files from the test description associated with the current result description.

/NOW

Concatenates all files selected for printing on the current PRINT command and immediately places them in the print queue.

The default is to place the concatenated files in the print queue when you exit from the Review subsystem.

/RESULT

Prints the result file from the specified result descriptions.

The default is */RESULT*.

/SELECTED

Concatenates all files already selected for printing and the currently specified file and immediately places them in the print queue.

The default is to place the concatenated files in the print queue when you exit from the Review subsystem.

/SUCCESSFUL

Prints files from result descriptions for tests that compared successfully.

The default is to select files from the test description associated with the current result description.

/UNSUCCESSFUL

Prints files from result descriptions for tests that compared unsuccessfully.

The default is to select files from the test description associated with the current result description.

/UPDATED

Prints files from result descriptions whose benchmark files have been updated.

The default is to select files from the test description associated with the current result description.

Command Parameter***result-description-expression***

Specifies one or more result descriptions from which the specified files are to be selected for printing. The result-description-name for a test is the same as its test name.

Description

The PRINT command selects one or more files for printing. DEC/Test Manager informs you if the specified files do not exist. When you exit from the Review subsystem, the files you select are concatenated and placed in the print queue as a single print job unless you specify otherwise. The PRINT/NOW command selects all files specified on the current print command, concatenates them, and places them immediately in the print queue. The PRINT/SELECTED command selects all files specified on all previous print commands during this review session as well as all files specified on the current print command, concatenates them, and places them immediately in the print queue.

Result and difference files you selected for printing, but subsequently deleted with an UPDATE command, are not printed. The EXIT/NOPRINT command specifies that the selected files are not to be submitted to the print queue.

Omitting the result-description-expression parameter causes DEC/Test Manager to select files from the current result description. Specifying a result-description-name causes DEC/Test Manager to select files from that result description only. That result description becomes the current result description. Specifying a result-description-expression containing wildcard characters causes DEC/Test Manager to select files from all result

DTM_REVIEW > PRINT

descriptions whose result-description-names match the result-description-expression. The current result description is not changed.

The comparison status qualifiers `/COMPARISON_ABORTED`, `/NEW`, `/NOT_RUN`, `/SUCCESSFUL`, `/UNSUCCESSFUL`, and `/UPDATED` specify that DEC/Test Manager select files from result descriptions with the specified comparison status. Including one or more comparison status qualifiers and a result-description-expression causes DEC/Test Manager to select files from all result descriptions that match both the result-description-expression and one of the qualifiers. The comparison status qualifiers cannot be included with a result-description-name.

The output file qualifiers `/BENCHMARK`, `/DIFFERENCE`, and `/RESULT` specify that DEC/Test Manager select the specified files for printing.

When DEC/Test Manager prints a benchmark file that is stored in a DEC/CMS library, the benchmark file is fetched and the fetched copy is deleted after it is printed.

DTM_REVIEW> SELECT

Moves you to the specified result description.

Format **DTM_REVIEW> SELECT** *result-description-name***Command Qualifiers**

None.

Defaults

None.

Command Parameter***result-description-name***

Specifies the name of the selected result description. The result-description-name for a test is the same as its test-name.

The result-description-name is required. You cannot specify a result-description-expression on the SELECT command.

Description

The SELECT command moves you to the specified result description. The specified result description becomes the current result description.

DTM_REVIEW > SHOW

DTM_REVIEW > SHOW

Describes and displays output files for specified result descriptions.

Format **DTM_REVIEW > SHOW** *[result-description-expression]*

Command Qualifiers

/BENCHMARK
/COMPARISON_ABORTED
/DIFFERENCE
/FILES
/NEW
/NOT_RUN
/OUTPUT[=file-spec]
/RESULT
/SUCCESSFUL
/SUMMARY
/UNSUCCESSFUL
/UPDATED

Defaults

/FILES
Current result description.
/FILES
/FILES
Current result description.
Current result description.
/OUTPUT=SYS\$OUTPUT
/FILES
Current result description.
/FILES
Current result description.
Current result description.

Command Qualifiers

/BENCHMARK

Displays the benchmark file. For a noninteractive test, DEC/Test Manager types the benchmark file. For an interactive test, DEC/Test Manager displays the benchmark file screen by screen and provides you with a keypad for manipulating the file. The keypad and the procedure for using it are described in Chapter 5.

If the benchmark file is in a DEC/CMS library, DEC/Test Manager fetches the benchmark element line by line and displays it on the screen.

If you include a comparison status qualifier with the */BENCHMARK* qualifier, DEC/Test Manager displays the benchmark file for result descriptions with the specified comparison status.

The default is to state whether output files (benchmark, difference, and result files) exist. If the benchmark file exists, its file specification is displayed.

/COMPARISON_ABORTED

States whether output files (benchmark, difference, and result files) exist for the specified result descriptions with the comparison aborted comparison status. If the benchmark file exists, its file specification is displayed.

If you include an output file qualifier with the */COMPARISON_ABORTED* qualifier, the specified output files are displayed for result descriptions with the comparison aborted comparison status. If the comparison for a test aborted, it has benchmark and result files. It might have a result file. You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

/DIFFERENCE

Displays the difference file. For a noninteractive test, DEC/Test Manager types the difference file. For an interactive test, DEC/Test Manager displays the benchmark and result files screen by screen with differences marked and provides you with a keypad for manipulating the files. The keypad and the procedure for using it are described in Chapter 5.

If you include a comparison status qualifier with the */DIFFERENCE* qualifier, DEC/Test Manager displays the difference file for result descriptions with the specified comparison status.

The default is to state whether output files (benchmark, difference, and result files) exist. If the benchmark file exists, its file specification is displayed.

/FILES

Displays the comparison status for result descriptions and states whether output files (benchmark, difference, and result files) exist for the result description. If the benchmark file exists, its file specification is displayed.

You cannot include the */FILES* qualifier on a command where you have included the */BENCHMARK*, */DIFFERENCE*, */RESULT*, or */SUMMARY* qualifiers.

The default is */FILES*.

/NEW

States whether output files (benchmark, difference, and result files) exist for result descriptions with the new comparison status.

DTM_REVIEW> SHOW

If you include an output file qualifier with the `/NEW` qualifier, the specified output files are displayed for result descriptions with the new comparison status. If a test is new, it has a result file but it does not have a benchmark or difference file. You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

/NOT_RUN

States whether output files (benchmark, difference, and result files) exist for result descriptions with the not run comparison status. If the benchmark file exists, its file specification is displayed.

If you include an output file qualifier with the `/NOT_RUN` qualifier, the specified output files are displayed for result descriptions with the not run comparison status. If a test does not run, it does not have a result or difference file. It might have a benchmark file. You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

/OUTPUT[=file-spec]

Sends the requested output to the specified file.

The default is `SYS$OUTPUT`.

/RESULT

Displays the result file. For a noninteractive test, DEC/Test Manager types the result file. For an interactive test, DEC/Test Manager displays the result file screen by screen and provides you with a keypad for manipulating the file. The keypad and the procedure for using it are described in Chapter 5.

If you include a comparison status qualifier with the `/RESULT` qualifier, DEC/Test Manager displays the result file for result descriptions with the specified comparison status.

The default is to state whether output files (benchmark, difference, and result files) exist. If the benchmark file exists, its file specification is displayed.

/SUCCESSFUL

States whether output files (benchmark, difference, and result files) exist for result descriptions with the successful comparison status. If the benchmark file exists, its file specification is displayed.

If you include an output file qualifier with the */SUCCESSFUL* qualifier, the specified output files are displayed for result descriptions with the successful comparison status. If a test is successful, it has a benchmark file. Its result file was deleted and no difference file was created. You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

/SUMMARY

Displays the Collection Summary Information (the information displayed when you first enter the Review subsystem). You cannot specify the */SUMMARY* and */FILES* qualifiers on the same command.

The */SUMMARY* qualifier is mutually exclusive with all qualifiers except the */OUTPUT* qualifier.

The default is */FILES*.

/UNSUCCESSFUL

States whether output files (benchmark, difference, and result files) exist for result descriptions with the unsuccessful comparison status. If the benchmark file exists, its file specification is displayed.

If you include an output file qualifier with the */UNSUCCESSFUL* qualifier, the specified output files are displayed for result descriptions with the unsuccessful comparison status. (If a test is unsuccessful, it has benchmark, result, and difference files.) You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

/UPDATED

States whether output files (benchmark, difference, and result files) exist for result descriptions with the updated comparison status. If the benchmark file exists, its file specification is displayed.

DTM_REVIEW > SHOW

If you include an output file qualifier with the /UPDATED qualifier, the specified output files are displayed for result descriptions with the updated comparison status. An updated test is a test whose benchmark file was created from its result file since the time when the test was last executed. An updated test does not have a result or difference file. You cannot specify a comparison status qualifier with a result-description-name parameter. A result-description-expression is required.

The default is to display information for the current result description.

Command Parameter

result-description-expression

Specifies one or more result descriptions about which the specified information is to be displayed. The result-description-name for a test is the same as its test-name.

Description

The SHOW command describes and displays output files for specified result descriptions.

Omitting the result-description-expression parameter displays information about the current result description. Specifying a result-description-name displays information about that result description only. That result description becomes the current result description. Specifying a result-description-expression containing wildcard characters displays information about each result description that matches the result-description-expression. The current result description does not change.

The comparison status qualifiers /COMPARISON_ABORTED, /NEW, /NOT_RUN, /SUCCESSFUL, /UNSUCCESSFUL, and /UPDATED can be specified only with a result-description-expression. They cannot be used with a result-description-name. Using comparison status qualifiers displays information about all result descriptions matching the result-description-expression and a comparison status qualifier.

The output file qualifiers /BENCHMARK, /DIFFERENCE, and /RESULT display files for the specified result descriptions or display a message indicating that the file does not exist. Using output file qualifiers with comparison status qualifiers displays the specified output files for result descriptions with the specified comparison status.

DTM_REVIEW > SHOW

The /FILES qualifier displays the comparison status for the specified result descriptions and, for each output file, states whether it exists. If a benchmark file exists, its file specification is also displayed. The /FILES qualifier cannot be specified with the output file qualifiers or with the /SUMMARY qualifier.

The /SUMMARY qualifier displays the Collection Summary Information. The /SUMMARY qualifier cannot be specified with the /FILES qualifier, with the output file qualifiers, or with the comparison status qualifiers.

DTM_REVIEW > SPAWN

DTM_REVIEW > SPAWN

Creates a subprocess of your current DEC/Test Manager session.

Format **DTM_REVIEW > SPAWN** *[command]*

Command Qualifiers

/CARRIAGE_CONTROL
/[NO]CLI[=cli]
/INPUT=file-spec
/[NO]KEYPAD
/[NO]LOGICAL_NAMES
/[NO]NOTIFY
/OUTPUT=file-spec
/PROCESS=subprocess-name
/[NO]PROMPT[=string]
/[NO]SYMBOLS
/[NO]WAIT

Defaults

See text.
See text.
See text.
/KEYPAD
/LOGICAL_NAMES
/NONOTIFY
See text.
See text.
See text.
/SYMBOLS
/WAIT

Command Qualifiers

/CARRIAGE_CONTROL
/NOCARRIAGE_CONTROL

Determines whether carriage control or line feed characters or both are prefixed to the prompt string of the subprocess.

The default is the current setting of the parent process.

/CLI[=cli]
/NOCLI

Specifies an alternate command language interpreter (CLI) for the subprocess to use. The CLI you specify must be located in SYS\$SYSTEM and have the file type EXE.

The default is the CLI the parent process uses.

/INPUT=file-spec

Specifies an input file containing one or more commands for the spawned subprocess to execute. If you specify a command with an input file, the command is processed before the commands in the input file. The subprocess terminates when processing is complete. You cannot use wildcards in the file specification.

/KEYPAD***/NOKEYPAD***

Determines whether DCL keypad symbols and the current DCL keypad state are copied from the DCL keypad in the parent process to the subprocess. Use the */NOKEYPAD* qualifier if you do not want the key settings to be copied.

The default is to copy any key definitions or states (or both) you have established with the *DEFINE/KEY* command.

/LOGICAL_NAMES***/NOLOGICAL_NAMES***

Determines whether the system passes process logical names and logical name tables to the subprocess.

The default is to copy all process logical names and logical name tables except those marked *CONFINE* or created in executive or kernel mode.

/NOTIFY***/NONOTIFY***

Determines whether a message is sent to your terminal to notify you that your subprocess has completed or aborted. Do not specify */NOTIFY* unless you also specify the */NOWAIT* qualifier.

The default is */NONOTIFY*.

/OUTPUT=file-spec

Specifies the output file to which the output of the SPAWN operation is to be written. When you specify */NOWAIT*, you should use */OUTPUT* to specify an output other than *SYS\$OUTPUT* to prevent your terminal from being used by both processes simultaneously.

The default is to direct the output to the current *SYS\$OUTPUT* device.

/PROCESS=subprocess-name

Specifies the name of the subprocess to be created.

DTM_REVIEW > SPAWN

The default is USERNAME_#. The pound sign (#) denotes a unique number.

/PROMPT[=string]

Specifies the DCL-prompt string for the subprocess. If you specify /PROMPT but do not specify a string, the default prompt is displayed.

The default is to copy the current prompt string from the parent process.

/SYMBOLS

/NOSYMBOLS

Determines whether the system passes DCL global and local symbols to the subprocess.

The default is /SYMBOLS.

/WAIT

/NOWAIT

Controls whether the system waits until the subprocess is completed before allowing more commands to be issued to the parent process.

The /NOWAIT qualifier allows you to enter more commands while the specified subprocess is running. When you specify /NOWAIT, you should also specify /OUTPUT to direct output to a file (rather than to the screen). This prevents your terminal from being used by both processes simultaneously.

The default is /WAIT.

Command Parameter

command

Specifies an optional command to be executed by the subprocess you are creating. If you specify the command parameter, you create a subprocess which executes the command and returns control to the DEC/Test Manager session when the command terminates. If you include the /INPUT qualifier with the command parameter, the subprocess reads commands from the specified input file after the command string executes. The command string cannot exceed 132 characters.

If you omit the command parameter, the SPAWN command creates a subprocess and attaches your terminal to it. You can return to your DEC/Test Manager session by logging out of the subprocess or by issuing the ATTACH/PARENT command. If you have created several subprocesses, you can switch between them using the ATTACH/IDENTIFICATION command.

Description

The SPAWN command creates a subprocess of your current DEC/Test Manager session (your parent process). The context of your DEC/Test Manager session is copied to the subprocess.

You can use the SPAWN command to leave the Review subsystem temporarily, to create another DEC/Test Manager session, or to edit a file, and then return to your original Review session.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). The benchmark file is updated even if you do not provide a remark. You must specify a result-description-expression in order to specify a remark.

Description

The UPDATE command creates a new benchmark file from the existing result file for the specified result descriptions. If the previous benchmark file is in the DEC/Test Manager library, it is deleted. To be updated, the result description must have a comparison status of comparison aborted, unsuccessful, or new. You cannot update the benchmark file for a result description with a comparison status of successful, not run, or updated. DEC/Test Manager automatically deletes the result file for successful tests.

Omitting the result-description-expression parameter causes DEC/Test Manager to update the benchmark file for the current result description, if its comparison status is comparison aborted, new, or unsuccessful. Specifying a result-description-name causes DEC/Test Manager to update the benchmark file for that result description only, providing that its comparison status is comparison aborted, new, or unsuccessful. This result description becomes the current result description. Specifying a result-description-expression containing wildcard characters causes DEC/Test Manager to update benchmark files for all test descriptions whose result-description-names match the result-description-expression and whose comparison status is comparison aborted, new, or unsuccessful. The current result description is not changed.

If you store your benchmark files in a DEC/CMS library, the UPDATE command issues the DEC/CMS RESERVE and REPLACE commands to replace the old benchmark file. The result file is used as input to the DEC/CMS REPLACE command. If you use DEC/CMS classes for your benchmark files, the UPDATE command also issues the DEC/CMS INSERT/SUPERSEDE command to place the current generation of the benchmark file into the class. If DEC/Test Manager is updating a generation other than the latest, it creates a variant line designated "D".

When you update a benchmark file in a DEC/CMS library for a new test with no existing benchmark file, DEC/Test Manager creates a new element in the specified DEC/CMS library. If you also specify a class, the new element is also inserted into the class.

DTM_REVIEW > UPDATE

Use the UPDATE command only when you are sure you want to delete the current benchmark file and replace it with the current result file. This procedure is irreversible; therefore, carefully consider the possible effects before using the UPDATE command.

You can also replace and delete benchmark files with the MODIFY TEST_DESCRIPTION/BENCHMARK and MODIFY TEST_DESCRIPTION/NOBENCHMARK commands.

DTM RUN

Executes a collection interactively.

Format **DTM RUN** *collection-name [remark]*

Command Qualifiers

/[NO]CONFIRM
/[NO]LOG
/[NO]LOG_FILE=file-spec
/[NO]OUTPUT=file-spec

Defaults

/CONFIRM
/LOG
See text.
/OUTPUT=SYS\$OUTPUT

Command Qualifiers

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts for confirmation when you rerun a collection.

The default is */CONFIRM*.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/LOG_FILE=file-spec

/NOLOG_FILE

Specifies whether a log file for the collection is to be created and where the contents of the log file are to be displayed. This file is similar in content to the log file created when you issue the DEC/Test Manager SUBMIT command to run a collection of tests in batch mode. The log file contains output the test run generates other than the output from the test itself, such as output the prologue and epilogue files generate.

The */LOG_FILE* qualifier creates a log file in the specified file. If you do not include the file-spec, the file is written to SYS\$OUTPUT.

DTM RUN

The default is `/LOG_FILE=SYS$OUTPUT`. If the `RUN` command is executed from a batch job, the default is `/NOLOG_FILE`.

/OUTPUT=file-spec

/NOOUTPUT

Specifies where a copy of the output of the `RUN` command (the test result files) is displayed or written.

The `/OUTPUT` qualifier directs the output to the specified file or device.

The `/NOOUTPUT` qualifier suppresses output.

Note that even when you enter the `/OUTPUT` or `/NOOUTPUT` qualifier to direct the output away from your terminal screen, your terminal is still tied up while the collection executes. It is recommended that you execute large collections in batch with the `SUBMIT` command, unless you need to see the output from your tests displayed.

By default, the output is directed to `SYS$OUTPUT`.

Command Parameters

collection-name

Identifies a collection created with the `CREATE COLLECTION` command.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (`"`).

Description

The `RUN` command executes a collection of tests interactively. Note that you cannot use the `DCL RUN` command to execute a collection of tests. The collection can contain interactive test descriptions, noninteractive test descriptions, or a combination of both. By default, the `RUN` command displays on your screen a copy of the information as it is put into the result file and any informational messages that are generated. This output can be redirected to a file with the `/OUTPUT` qualifier or suppressed with the `/NOOUTPUT` qualifier. Note that using either of these qualifiers also ties up your terminal while the collection executes. Therefore, when you do not need to see the output from your test, it is recommended that you execute collections in batch using the `DTM SUBMIT` command.

If you want to execute a collection in batch, use the SUBMIT command rather than submitting a command procedure that includes the RUN command. If you do plan to execute the RUN command from batch, you must specify /NOLOG_FILE or /LOG_FILE=file-spec, where file-spec specifies something other than SYS\$OUTPUT. If /LOG_FILE specifies SYS\$OUTPUT, the RUN command will attempt to write to your batch job's log file. The VAX/VMS operating system will not allow this file to be shared, and your RUN command will fail.

If you specify a test-name rather than a collection-name as the parameter to the RUN command, DEC/Test Manager prompts you to confirm that you want to create a collection containing the test. If you type YES, DEC/Test Manager creates a collection with the same collection-name as the specified test-name—if a collection by that name does not already exist—and executes this collection.

Press CTRL/C to abort a RUN command. Pressing CTRL/C allows DEC/Test Manager to clean up and restore the library to a consistent state. Do not press CTRL/Y to abort a collection run.

Example

```
DTM> RUN TIME "simple test of TIME command"
Starting TIME test run...
%DTM-I-BEGIN, your interactive test session is now beginning...
.
.
%DTM-S-CONCLUDED, your interactive test session has concluded
Performing post-run cleanup with comparison...
DTM-I-SUCCEEDED, the comparison for test TIME succeeded
DTM-S-COMPARED, collection TIME compared
DTM>
```

This command executes the collection TIME interactively.

DTM SET BENCHMARK_DIRECTORY

DTM SET BENCHMARK_DIRECTORY

Specifies the default directory that DEC/Test Manager searches for benchmark files.

Formats	DTM SET BENCHMARK_DIRECTORY <i>directory-spec</i> <i>[remark]</i>
	DTM SET NOBENCHMARK_DIRECTORY <i>[remark]</i>
Command Qualifier	Default
/[NO]LOG	/LOG

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

directory-spec

Specifies the directory that DEC/Test Manager searches by default for benchmark files. Unless instructed otherwise, DEC/Test Manager places updated benchmark files in this directory. This directory may be a DEC/CMS library.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (""). If you do not provide a remark, you are prompted for one, but DEC/Test Manager establishes or cancels the directory even if you do not provide a remark.

DTM SET BENCHMARK_DIRECTORY

Description

The SET BENCHMARK_DIRECTORY command instructs DEC/Test Manager to establish the specified directory as the default directory for benchmark files for the current DEC/Test Manager library. By default, DEC/Test Manager searches this directory for benchmark files and places new and updated benchmark files in this directory.

When you create a collection with the CREATE COLLECTION/BENCHMARK_DIRECTORY command and specify a default benchmark directory for the collection, DEC/Test Manager references that directory for benchmark files associated with tests in that collection. If you associate a directory specification with the benchmark file for a specific test, DEC/Test Manager references that directory rather than either the default benchmark directory or the collection benchmark directory when executing that test.

If the directory you specify is a DEC/CMS library, DEC/Test Manager issues the appropriate DEC/CMS commands to remove, replace, or insert files in the library.

If you store your benchmark files in a DEC/CMS library, you can use classes to specify which version of the benchmark file you want DEC/Test Manager to access. This allows you to select from several versions of a single benchmark file, which is useful if you are maintaining several base levels of a program.

The SET NOBENCHMARK_DIRECTORY command clears any library default benchmark directory established with the SET BENCHMARK_DIRECTORY command.

Example

```
DTM> SET BENCHMARK_DIRECTORY DRA1:[projectlib.bmk]
_Remark: New default benchmark directory
%DTM-S-NEWDEF, DRA1:[projectlib.bmk] is the new default collection benchmark directory
DTM>
```

This command establishes a new default benchmark directory for the library.

DTM SET EPILOGUE

DTM SET EPILOGUE

Establishes the specified file as the default epilogue for all subsequently created collections.

Formats **DTM SET EPILOGUE** *file-spec [remark]*
DTM SET NOEPILOGUE *[remark]*

Command Qualifier	Default
/[NO]LOG	/LOG

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

file-spec

Specifies the file as the default collection epilogue.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager establishes or cancels the collection epilogue file even if you do not provide a remark.

Description

The SET EPILOGUE command establishes the default epilogue file for subsequently created collections. When you establish the default epilogue file, DEC/Test Manager does not check to see if the file exists. This check is done when a collection is created. This file may be in a DEC/CMS library.

You can override the default when you create a collection by assigning another file with the CREATE COLLECTION/EPILOGUE command.

The SET NOEPILOGUE command cancels a previously established default collection epilogue file.

Examples

1.

```
⌘ DTM SET EPILOGUE DRA1:[project.lib]EPILOGUE.COM ""
%DTM-S-NEWDEF, DRA1:[project.lib]epilogue.com is the new default
collection epilogue
```

This example specifies a file as the default epilogue file for subsequently created collections.

2.

```
⌘ DTM SET NOEPILOGUE "canceling epilogue"
DTM-S-DEFCANCEL, default epilogue canceled
```

This example cancels a default collection epilogue file.

DTM SET LIBRARY

DTM SET LIBRARY

Selects an existing DEC/Test Manager library.

Format **DTM SET LIBRARY** *directory-spec*

Command Qualifiers

/[NO]VERIFY

/[NO]LOG

Defaults

/VERIFY

/LOG

Command Qualifiers

/VERIFY

/NOVERIFY

Specifies whether DEC/Test Manager is to verify that the specified directory is a valid DEC/Test Manager library.

The */VERIFY* qualifier causes DEC/Test Manager to verify that the specified directory is a valid DEC/Test Manager library before selecting it as the current DEC/Test Manager library. If the specified directory is not a valid library, DEC/Test Manager issues an error message.

The */NOVERIFY* qualifier causes DEC/Test Manager to select the library without verifying that it is a valid DEC/Test Manager library. You must specify a valid DEC/Test Manager library.

By default, DEC/Test Manager verifies that the directory is valid before selecting it as the current library.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameter

directory-spec

Specifies a directory that has been defined as a DEC/Test Manager library with the CREATE LIBRARY command. The file-spec you specify cannot be your current default directory, even if this directory is a valid DEC/Test Manager library.

Description

The SET LIBRARY command selects an existing DEC/Test Manager library for use with DEC/Test Manager commands. This library is your current library and remains your current library until you select another library with the CREATE LIBRARY or SET LIBRARY command, or until you log out.

An error message is issued if the SET LIBRARY command is used with a directory specification that is not a DEC/Test Manager library.

Example

```
⌘ DTM SET LIBRARY [ .DTMLIB ]  
%DTM-S-LIBIS, DEC/Test Manager library is DISK⌘USER01:[PROJECT.DTMLIB]
```

This example selects a DEC/Test Manager library.

DTM SET PROLOGUE

DTM SET PROLOGUE

Establishes the specified file as the default prologue file for all subsequently created collections.

Formats	DTM SET PROLOGUE <i>file-spec [remark]</i>
	DTM SET NOPROLOGUE <i>[remark]</i>
Command Qualifier	Default
/[NO]LOG	/LOG

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

file-spec

Specifies the file to be the default collection prologue file.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (""). If you do not provide a remark, you are prompted for one, but DEC/Test Manager establishes or cancels the collection prologue file even if you do not provide a remark.

Description

The SET PROLOGUE command establishes the specified file as the default prologue file for subsequently created collections. When you establish the default prologue file, DEC/Test Manager does not check to see if the file exists. This check is done when a collection is created. This file may be in a DEC/CMS library.

You can override the default when you create a collection by assigning another file with the CREATE COLLECTION/PROLOGUE command.

The SET NOPROLOGUE command cancels a previously established default collection prologue file.

Examples

1.

```
$ DTM SET PROLOGUE DRA1:[project.lib]PROLOGUE.COM ""  
%DTM-S-NEWDEF, DRA1:[project.lib]prologue.com is the new default  
collection prologue.
```

This example specifies a default collection prologue file.

2.

```
$ DTM SET NOPROLOGUE "canceling prologue"  
DTM-S-DEFCANCEL, Default prologue canceled
```

This example cancels a default collection prologue file.

DTM SET TEMPLATE_DIRECTORY

DTM SET TEMPLATE_DIRECTORY

Establishes the default directory that DEC/Test Manager searches for test template files.

Formats **DTM SET TEMPLATE_DIRECTORY** *directory-spec* [*remark*]
DTM SET NOTEMPLATE_DIRECTORY [*remark*]

Command Qualifier	Default
/[NO]LOG	/LOG

Command Qualifier

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

Command Parameters

directory-spec

Specifies the directory that DEC/Test Manager searches by default for template files. Unless otherwise instructed, DEC/Test Manager uses this directory. This directory may be a DEC/CMS library.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (" "). If you do not provide a remark, you are prompted for one, but DEC/Test Manager establishes or cancels the default template directory even if you do not provide a remark.

DTM SET TEMPLATE_DIRECTORY

Description

The SET TEMPLATE_DIRECTORY command instructs DEC/Test Manager to establish the specified directory as the default directory for template files in the current DEC/Test Manager library. By default, DEC/Test Manager searches this directory for template files. When you create a collection with the CREATE COLLECTION /TEMPLATE_DIRECTORY command and specify a default template directory for the collection, DEC/Test Manager references that directory for templates associated with tests in that collection. If you include a directory specification in the template file specification for a test in the collection, DEC/Test Manager references that directory rather than either the default template directory or the collection template directory when executing that test.

If the directory you specify is a DEC/CMS library, DEC/Test Manager issues the appropriate DEC/CMS commands to remove, replace, or insert files in the library. These DEC/CMS commands are logged in the DEC/Test Manager history file.

If you store your template files in a DEC/CMS library, you can use classes to specify which version of the template file you want DEC/Test Manager to access.

The SET NOTEMPLATE_DIRECTORY command clears any library default template directory established with the SET BENCHMARK_DIRECTORY command.

Example

```
DTM> SET TEMPLATE_DIRECTORY DRA1:[projectlib.tpl]
_Remark: New default template directory
%DTM-S-NEWDEF, DRA1:[projectlib.tpl] is the new default collection template directory
DTM>
```

This command establishes a new default template directory for the library.

DTM SHOW ALL

DTM SHOW ALL

Displays a summary describing the current library.

Format

DTM SHOW ALL

Command Qualifier

/OUTPUT[=*file-spec*]

Default

/OUTPUT=SYS\$OUTPUT

Command Qualifier

*/OUTPUT[=*file-spec*]*

Sends the SHOW ALL information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW ALL command displays the current directory specifications for the directories or DEC/CMS libraries containing your epilogue, prologue, benchmark and template files, and the number of collections, test descriptions, groups, and variables in the library.

Example

```
DTM> SHOW ALL
```

```
Description of DEC/Test Manager Library DBA1$: [project.dsrlib]
```

```
Default template directory: DBA1$: [projectlib.tpl] "Default template library"
```

```
Default benchmark directory: DBA1$: [projectlib.bmk] "Default benchmark library"
```

```
Default collection prologue: DBA1$: [project.mylib]PROLOGUE_1.COM
```

```
Default collection epilogue: DBA1$: [project.mylib]EPILOGUE_1.COM
```

```
Number of collections: 25
```

```
Number of test descriptions: 48
```

```
Number of groups: 8
```

```
Number of variables: 7
```

```
DTM>
```

This example describes the current DEC/Test Manager library.

DTM SHOW BENCHMARK_DIRECTORY

DTM SHOW BENCHMARK_DIRECTORY

Displays the directory specification for the current default benchmark directory.

Format DTM SHOW BENCHMARK_DIRECTORY

Command Qualifier
/OUTPUT[=file-spec]

Default
/OUTPUT=SYS\$OUTPUT

Command Qualifier

/OUTPUT[=file-spec]

Sends the SHOW BENCHMARK_DIRECTORY information to a specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW BENCHMARK_DIRECTORY command displays the directory specification for the default directory for benchmark files. This directory can be a DEC/CMS library.

Example

```
DTM> SHOW BENCHMARK_DIRECTORY
DBA1$: [projectlib.bmk]           "Default benchmark library"
DTM>
```

This command displays the directory specification for the current default benchmark directory.

DTM SHOW COLLECTION

DTM SHOW COLLECTION

Displays information about collections in the current DEC/Test Manager library.

Format **DTM SHOW COLLECTION** *collection-expression*

Command Qualifiers

Defaults
/BENCHMARK_DIRECTORY
/BRIEF
/CLASS=(keyword=class-name,...)
/FULL
/INTERMEDIATE
/OUTPUT[=file-spec]
/TEMPLATE_DIRECTORY

None.
/INTERMEDIATE
None.
/INTERMEDIATE
/INTERMEDIATE
/OUTPUT=SYS\$OUTPUT
None.

Command Qualifiers

/BENCHMARK_DIRECTORY

Displays the directory specification of the benchmark directory used by this collection.

/BRIEF

Displays the collection-name.

The default is /INTERMEDIATE.

/CLASS=(keyword=class-name,...)

Displays the optional DEC/CMS class for benchmark and template files stored in DEC/CMS libraries. The keywords are BENCHMARK and TEMPLATE. The class-names you specify can be the same, or you can use different class-names for your benchmark and template files.

If you specify both keywords, separate them with a comma and enclose the list in parentheses. If you specify only one keyword, you can omit the parentheses. If you do not specify a keyword, both classes are displayed.

/FULL

Displays the following output:

Collections in the DEC/Test Manager Library library-name

```
COLLECTION NAME  NUMBER tests  DATE  TIME "remark"
Command: COMMAND-LINE
Status: RUN, COMPARISON, AND REVIEW STATUS
Successful count: NUMBER      Unsuccessful count: NUMBER
New test count: NUMBER        Updated test count: NUMBER
Test not run count: NUMBER    Comparisons aborted: NUMBER
Default template directory: DIRECTORY_SPEC
Template class: CLASS_NAME
Default benchmark directory: DIRECTORY_SPEC
Benchmark class: CLASS_NAME
Prologue: FILE_SPEC
Epilogue: FILE_SPEC
Last Review: DATE TIME
```

A result description for each test in the collection follows this display. A result description lists the test-name and its comparison status and the file specifications for the benchmark, result, and difference files.

The default is */INTERMEDIATE*.

/INTERMEDIATE

Displays the following output:

Collections in the DEC/Test Manager library library-name

```
COLLECTION NAME  NUMBER tests  DATE  TIME "remark"
Command: COMMAND-LINE
Status: RUN, COMPARISON, AND REVIEW STATUS
Successful count: NUMBER      Unsuccessful count: NUMBER
New test count: NUMBER        Updated test count: NUMBER
Test not run count: NUMBER    Comparisons aborted: NUMBER
```

The default is */INTERMEDIATE*.

/OUTPUT[=file-spec]

Sends the SHOW COLLECTION information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

/TEMPLATE_DIRECTORY

Displays the directory specification of the template directory used by this collection. Displays the directory specification of the default directory for the current library.

DTM SHOW COLLECTION

Command Parameter

collection-expression

Specifies the collections about which information is to be displayed. A collection-expression can be a collection-name, a wildcard, a wildcard in combination with a collection-name, or a list of these separated by commas.

Description

The SHOW COLLECTION command displays information about collections in the current DEC/Test Manager library. These collections include currently running collections, collections that have been created but not run, collections that have been run but not deleted, and collections that have been run and stopped (partially run collections), but not deleted.

The /BRIEF, /INTERMEDIATE, and /FULL qualifiers determine how much information is displayed. The /BENCHMARK_DIRECTORY, /CLASS, and /TEMPLATE_DIRECTORY qualifiers add the specified additional information to the display. If an item does not exist in the collection, DEC/Test Manager displays "None specified" in the appropriate field.

Example

```
DTM> SHOW COLLECTION/BRIEF
Collections in DEC/Test Manager Library DBA1$: [project.dsrlib]
RNOCOLLECT
DTM>
```

This command lists the names of all collections in the current library.

DTM SHOW EPILOGUE

Displays the file specification for the default epilogue for collections.

Format DTM SHOW EPILOGUE

Command Qualifier
/OUTPUT[=*file-spec*]

Default
/OUTPUT=SYS\$OUTPUT

Command Qualifier

*/OUTPUT[=**file-spec**]*

Sends the SHOW EPILOGUE information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW EPILOGUE command displays the file specification for the default epilogue for collections. The default collection epilogue is established with the SET EPILOGUE command.

Example

```
DTM> SHOW EPILOGUE
DBA1$: [project.mylib]EPILOGUE_1.COM "Default collection epilogue file"
DTM>
```

This command displays the file specification for the default collection epilogue file.

DTM SHOW GROUP

DTM SHOW GROUP

Displays the contents of one or more groups.

Format	DTM SHOW GROUP	<i>group-expression</i>
	Command Qualifiers	Defaults
	/BRIEF	/INTERMEDIATE
	/[NO]CONTENTS[=n]	/CONTENTS=1
	/FULL	/INTERMEDIATE
	/INTERMEDIATE	/INTERMEDIATE
	/MEMBER	None.
	/OUTPUT[=file-spec]	/OUTPUT=SYS\$OUTPUT

Command Qualifiers

/BRIEF

Lists the group-name only. The format is

group-name

You cannot specify both the /CONTENTS and /BRIEF qualifiers on the same command.

The default is /INTERMEDIATE.

/CONTENTS[=n]

/CONTENTS[=ALL]

/NOCONTENTS

Displays the name of each group and the names of all test descriptions and groups that are contained in the specified groups. You cannot specify the /CONTENTS and /BRIEF qualifiers on the same command. You cannot specify the /CONTENTS, /FULL, and /MEMBER qualifiers on the same command.

The /CONTENTS=n qualifier displays this information for n levels of groups within each specified group. If n is greater than one, all nested groups for n levels are expanded and displayed.

The /CONTENTS=ALL qualifier displays this information for all groups within each group.

DTM SHOW GROUP

The `/NOCONTENTS` qualifier displays the group-names only.

The default is `/CONTENTS=1`.

/FULL

Displays the same output as the `/CONTENTS=ALL` qualifier. You cannot specify the `/CONTENTS`, `/FULL`, and `/MEMBER` qualifiers on the same command.

The default is `/INTERMEDIATE`.

/INTERMEDIATE

Lists the group-name and any remark added during the creation, modification, or copying of the group. The format is

```
group-name    "remark"
```

The default is `/INTERMEDIATE`.

/MEMBER

Lists the groups of which the specified groups are members. The format of the output is

```
GROUP NAME    "REMARK"
```

Is a member of the following groups:

```
GROUP NAME  
GROUP NAME  
GROUP NAME
```

You cannot specify the `/CONTENTS`, `/FULL`, and `/MEMBER` qualifiers on the same command.

/OUTPUT[=file-spec]

Sends the SHOW GROUP information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is `SY$OUTPUT`.

Command Parameter

group-expression

Specifies the groups about which information is to be displayed. A group-expression can be a group-name, a wildcard, a wildcard in combination with a group-name, or a list of these separated by commas.

DTM SHOW GROUP

Description

The SHOW GROUP command lists a group's name, its contents, and its creation remark. If you specify the SHOW GROUP command for more than one group, the groups are described in alphabetical order.

If you omit the group-expression parameter, DEC/Test Manager displays information about all groups.

Example

```
$ DTM SHOW GROUP/CONTENTS=ALL MARGINS
Groups in DEC/Test Manager Library DISK$USER01: [PROJECT.DTMLIB]

MARGINS "test of margin commands"
  LEFTMARGINS /GROUP
    LMTEST1
    LMTEST2
  RIGHTMARGINS /GROUP
    RMTEST1
    RMTEST2
  TEST1
```

This example displays the contents of the MARGINS group. Because /CONTENTS=ALL is specified, each group, test description, and remark is listed.

DTM SHOW HISTORY

Displays a chronological list of the transactions performed on your DEC/Test Manager library.

Format **DTM SHOW HISTORY** [*object-expression*]

Command Qualifiers

/BEFORE=time
 /OUTPUT=file-spec
 /SINCE=time
 /[NO]TRANSACTIONS=(keyword,...)
 /USER=user-name

Defaults

See text.
 /OUTPUT=SYS\$OUTPUT
 Library creation.
 /TRANSACTIONS=ALL
 None.

Command Qualifiers

/BEFORE=time

Lists all history information prior to a specified date. By default, the time is the current date and time. The time value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY.

The default is to display information prior to the time you enter the command.

/OUTPUT=file-spec

Sends the SHOW HISTORY information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is /OUTPUT=SYS\$OUTPUT.

/SINCE=time

Specifies that only those history entries dated on or after the given time are to be displayed. The time value can be an absolute, delta, or combination time value, or one of the following keywords: TODAY, TOMORROW, or YESTERDAY. If you specify /SINCE but do not specify a value, DEC/Test Manager defaults to /SINCE=TODAY.

By default, all transactions recorded since the library was created are displayed.

DTM SHOW HISTORY

/TRANSACTION=(keyword,...)

/NOTTRANSACTION=(keyword,...)

Displays all transaction records generated by the commands associated with the keywords you specify. The valid keyword values are as follows:

ALL	Displays all transactions recorded in the history file
COPY	Displays all COPY commands
CREATE	Displays all CREATE commands
DELETE	Displays all DELETE commands
INSERT	Displays all INSERT commands
MODIFY	Displays all MODIFY commands
RECREATE	Displays all RECREATE commands.
REMARK	Displays all REMARK commands
REMOVE	Displays all REMOVE commands
REVIEW	Displays all REVIEW commands
RUN	Displays all RUN commands
SET	Displays all SET commands
SUBMIT	Displays all SUBMIT commands
STOP	Displays all STOP commands
UPDATE	Displays all Review UPDATE commands
VERIFY	Displays all VERIFY commands

If you specify more than one command, you must enclose the list of commands in parentheses and separate them with commas.

The */TRANSACTION* qualifier directs DEC/Test Manager to list transaction records only for the listed commands.

The */NOTTRANSACTION* qualifier directs DEC/Test Manager to list transaction records for all commands except the listed commands.

The default is to list transaction records for all transactions in the DEC/Test Manager library.

/USER=user-name

Lists in chronological order the library transactions performed by the specified user. You cannot use wildcards in user names.

Command Parameter

object-expression

Specifies a DEC/Test Manager object about which history information is to be displayed. The object-expression can be a test-name, a group-name, or a collection-name; a wildcard test-name, group-name, or collection-name; or a list of these names separated by commas.

Description

The SHOW HISTORY command displays a chronological list of transactions performed on your DEC/Test Manager library. If you specify an object-expression, history information about that DEC/Test Manager entity is displayed. DEC/Test Manager records all transactions that alter the library. The transactions performed with the following DEC/Test Manager commands are logged in the history file:

COPY	CREATE	DELETE
INSERT	MODIFY	RECREATE
REMARK	REMOVE	REVIEW
RUN	SET	SUBMIT
STOP	UPDATE	VERIFY

The SHOW HISTORY command qualifiers determine which library transactions should be reported. The qualifiers you use specify all the conditions that must be true for a particular transaction record to be printed.

The SHOW HISTORY command displays the transaction records in the following format:

date time user command remark

One space separates each item in the transaction record. Each item is explained in the following section.

date

Displays the date the command was issued, in the following format:

[d]d-mmm-yyyy

For example:

8-JAN-1985

DTM SHOW HISTORY

time

Displays the time the command was issued, in the following format:

hh:mm:ss

For example:

15:10:12

user

Displays the user name of the person who issued the command.

command

Identifies the command that was issued. The string "DTM" does not appear as part of the command. The subcommand name, option name (if any), and any parameters are displayed. If the command operates on a particular generation of an element, the element name is followed by the generation number enclosed in parentheses, for example, PROLOGUE.COM(3). Qualifiers that indicate some modification of the library are logged, for example, the /RECOVER qualifier on the VERIFY command.

remark

Displays the remark entered with the command. The remark is enclosed in quotation marks ("). If no remark is entered, a null remark (" ") is displayed.

The following is an example transaction record:

```
06-APR-86 17:15:31 SMITH COMPARE lncollect1 "compare the simple left  
margin tests"
```

This record shows that on April 6, 1986, at 5:15 PM, SMITH issued the DTM COMPARE command for the collection of tests LMCOLLECT1.

For any command that causes an unusual occurrence, DEC/Test Manager displays an asterisk (*) in the first column.

Note that since transaction records for tests or groups deleted from the library are retained, SHOW HISTORY can display records for tests and groups that do not currently exist. If you reuse a deleted name, SHOW HISTORY does not distinguish between the old and new histories.

Example

```
⌘ DTM SHOW HISTORY
History in DEC/TEST MANAGER Library DRA1:[project.dtmlib]
08-JAN-86 15:33:59 SMITH CREATE LIBRARY DRA1:[PROJECT.DTMLIB] "New
DEC/Test Manager library"
08-JAN-86 15:55:45 SMITH CREATE TEST_DESCRIPTION LMTEST1 "Simple
test of left margin command"
08-JAN-86 16:15:35 SMITH CREATE TEST_DESCRIPTION/RECORD MEMO_TEST "First
interactive test of memo template"
.
.
.
⌘
```

This command displays the history file for the current library.

DTM SHOW LIBRARY

DTM SHOW LIBRARY

Displays the directory specification of the current DEC/Test Manager library.

Format

DTM SHOW LIBRARY

Command Qualifier	Default
-------------------	---------

<code>/OUTPUT[=file-spec]</code>	<code>/OUTPUT=SYS\$OUTPUT</code>
----------------------------------	----------------------------------

Command Qualifier

`/OUTPUT[=file-spec]`

Sends the SHOW LIBRARY information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW LIBRARY command displays the file specification of the current DEC/Test Manager library, or informs you if the library is invalid or if no library has been set. If the library is invalid, DEC/Test Manager issues an error message instructing you to use the VERIFY command.

Example

```
⌘ DTM SHOW LIBRARY
Your DEC/Test Manager library is DISK$USER01:[PROJECT.DTMLIB]
```

This example displays your current DEC/Test Manager library.

DTM SHOW PROLOGUE

Displays the file specification of the default prologue for collections.

Format DTM SHOW PROLOGUE

Command Qualifier

/OUTPUT[=file-spec]

Default

/OUTPUT=SYS\$OUTPUT

Command Qualifier

/OUTPUT[=file-spec]

Sends the SHOW PROLOGUE information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW PROLOGUE command displays the file specification for the default prologue for collections. This is established with a previous SET PROLOGUE command.

Example

```
DTM> SHOW PROLOGUE
DBA1$: [project.mylib]PROLOGUE_1.COM  "Default collection prologue file"
DTM>
```

This command displays the file specification for the default collection prologue file.

DTM SHOW TEMPLATE_DIRECTORY

DTM SHOW TEMPLATE_DIRECTORY

Displays the directory specification for the default directory for template files.

Format DTM SHOW TEMPLATE_DIRECTORY

Command Qualifier

/OUTPUT[=file-spec]

Default

/OUTPUT=SYS\$OUTPUT

Command Qualifier

/OUTPUT[=file-spec]

Sends the SHOW TEMPLATE_DIRECTORY information to a specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW TEMPLATE_DIRECTORY command displays the directory specification for the default directory for template files. This directory can be a DEC/CMS library.

Example

```
DTM> SHOW TEMPLATE_DIRECTORY
DBA1$: [projectlib.tmpl]           "Default template library"
DTM>
```

This command displays the directory specification for the current default template directory.

DTM SHOW TEST_DESCRIPTION

Displays the files and groups associated with one or more test descriptions.

Format **DTM SHOW TEST_DESCRIPTION** *test-group-expression*

Command Qualifiers

/BENCHMARK
/BRIEF
/COMPARISON_TYPE
/EPILOGUE
/FILTER
/FULL
/GROUPS
/INTERMEDIATE
/OUTPUT[=file-spec]
/PROLOGUE
/REMARK
/TEMPLATE
/TYPE
/VARIABLE

Defaults

See text.
/INTERMEDIATE
See text.
See text.
See text.
/INTERMEDIATE
See text.
/INTERMEDIATE
/OUTPUT=SYS\$OUTPUT
See text.
See text.
See text.
See text.
See text.

Parameter Qualifiers

/GROUP
/TEST_DESCRIPTION

Defaults

/TEST_DESCRIPTION
/TEST_DESCRIPTION

Command Qualifiers

/BENCHMARK

Displays the file specification for the benchmark file associated with the specified test description if you specified a benchmark on the CREATE TEST_DESCRIPTION command.

The following is displayed if the file exists and if you use the default benchmark when you create the test description:

`test-name.BMK`

DTM SHOW TEST_DESCRIPTION

/BRIEF

Displays the test-name.

The default is */INTERMEDIATE*.

/COMPARISON_TYPE

Displays the comparison type associated with the specified test description if you specified a comparison type on the CREATE or MODIFY TEST_DESCRIPTION command.

/EPILOGUE

Displays the file specification for the epilogue file associated with the specified test description.

/FILTER

Displays the names for all filters associated with the specified test description.

/FULL

Displays the contents of all test description fields. The format is:

Test descriptions in DEC/Test Manager library library-name

Test_name	"remark"
Template	= file-name
Benchmark	= file-name
Prologue	= file-name
Epilogue	= file-name
Variables	= variable-name(s) [=value]
Groups	= group-name(s)
Type	= type-name
Filters	= filter-type(s)
Comparison Type	= comparison-type

The default is */INTERMEDIATE*.

/GROUPS

Displays the names of the groups with which the test is affiliated. You cannot abbreviate this qualifier. */GROUPS* is the shortest unique form of this qualifier because of the existence of the */GROUP* parameter qualifier for this command.

DTM SHOW TEST_DESCRIPTION

/INTERMEDIATE

Displays the contents of the benchmark, template, prologue, and epilogue test description fields. The format is

Test descriptions in DEC/Test Manager library library-name

```
Test_name      "remark"  
Template = file-name  
Benchmark= file-name  
Prologue = file-name  
Epilogue = file-name
```

The default is */INTERMEDIATE*.

/OUTPUT[=file-spec]

Sends the SHOW TEST_DESCRIPTION information to a specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

/PROLOGUE

Displays the file specification for the prologue file associated with the specified test description.

/REMARK

Displays the remark associated with the test description.

/TEMPLATE

Displays the file specification for the template associated with the specified test description.

/TYPE

Displays the type of the specified test description—INTERACTIVE or NONINTERACTIVE.

/VARIABLE

Displays the names and values for all variables associated with the test description.

DTM SHOW TEST_DESCRIPTION

Parameter Qualifiers

/GROUP

Identifies the immediately preceding item in the parameter as a group-expression. If a test-group-expression comprises a list, use this qualifier after each item that designates a group.

You cannot abbreviate this qualifier. /GROUP is the shortest unique form of the parameter qualifier because of the existence of the /GROUPS qualifier for this command.

The default is /TEST_DESCRIPTION.

/TEST_DESCRIPTION

Identifies the immediately preceding item in the parameter as a test-expression.

The default is /TEST_DESCRIPTION.

Command Parameter

test-group-expression

Specifies tests or groups of tests about which to display information. A test-expression can be a test-name or a group-name, wildcard forms of these, or a list of these separated by commas.

Identify items in a test-expression as test or group items with the /GROUP or /TEST_DESCRIPTION parameter qualifier.

Description

The SHOW TEST_DESCRIPTION command displays files and groups associated with the specified test descriptions. The /BENCHMARK, /COMPARISON_TYPE, /FILTER, /EPILOGUE, /PROLOGUE, /TEMPLATE, and /VARIABLE qualifiers display the contents of the specified test description fields. The /GROUPS, /REMARK, and /TYPE qualifiers display the groups and remark associated with the test description, and whether the test description describes an interactive or noninteractive test. The /BRIEF, /INTERMEDIATE, and /FULL qualifiers limit the amount of information displayed. The /OUTPUT qualifier redirects the output from the command.

If you do not supply a test expression, DEC/Test Manager displays the intermediate level of information for all tests in the library by test-name.

Example

```
⌘ DTM SHOW TEST_DESCRIPTION LMTEST1
Test descriptions in DEC/Test Manager Library DISK$USER01:[PROJECT.DTMLIB]
LMTEST1          "left margin test"
Template = LMTEST1.RNO
Benchmark= BENCHLIB=LMTEST1.BMK
Prologue = None Specified
Epilogue = None Specified
```

This example displays the contents of the test description LMTEST1.

DTM SHOW VARIABLE

DTM SHOW VARIABLE

Displays information about the specified variables.

Format	DTM SHOW VARIABLE	<i>variable-expression</i>
	Command Qualifiers	Defaults
	/BRIEF	/INTERMEDIATE
	/FULL	/INTERMEDIATE
	/INTERMEDIATE	/INTERMEDIATE
	/OUTPUT[= <i>file-spec</i>]	/OUTPUT=SYS\$OUTPUT
	/REMARK	See text.
	/SCOPE	See text.
	/TEST_DESCRIPTION	See text.
	/TYPE	See text.
	/USAGE	See text.
	/VALUE	See text.

Command Qualifiers

/BRIEF

Displays the variable-name only.

The default is /INTERMEDIATE.

/FULL

Displays the name, value, scope, usage, and remark for the variable and lists the tests with which the variable is associated.

The default is /INTERMEDIATE.

/INTERMEDIATE

Displays the name, value, and remark for the variable.

The default is /INTERMEDIATE.

/OUTPUT[=*file-spec*]

Sends the SHOW VARIABLE information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

DTM SHOW VARIABLE

The default is SYS\$OUTPUT.

/REMARK

Displays the remark associated with the variable.

/SCOPE

Displays the variable's scope. The scope can be either global or local.

/TEST_DESCRIPTION

Lists the test descriptions with which the variable is associated.

/TYPE

Displays the variable's type.

/USAGE

Displays the variable's usage. The usage can be either as a symbol or as a logical.

/VALUE

Displays the variable's default value. If no value has been supplied, the null value is displayed.

Command Parameter

variable-expression

Specifies the variable about which to display information. A variable-expression can be a variable-name, a wildcard, a wildcard in combination with a variable-name, or a list of these separated by commas.

Description

The SHOW VARIABLE command displays information about the specified variables. The variable must be previously defined in the DEC/Test Manager library with the CREATE VARIABLE command.

The default qualifier, */INTERMEDIATE*, displays the name, value, and remark for a variable.

DTM SHOW VARIABLE

Example

```
⌘ DTM SHOW VARIABLE /VALUE INPUT_FILE
```

Variables in the DEC/Test Manager library library-name

```
INPUT_FILE    "input file for DSR tests"  
Value = INPUT.RNO
```

This example displays the value of the variable INPUT_FILE.

DTM SHOW VERSION

Displays the version number for DEC/Test Manager currently in use on the system.

Format **DTM SHOW VERSION**

Command Qualifier
/OUTPUT[=file-spec]

Default
/OUTPUT=SYS\$OUTPUT

Command Qualifier

/OUTPUT[=file-spec]

Sends information to the specified file. If you specify the file-name but omit the file-type, the file-type defaults to LIS.

The default is SYS\$OUTPUT.

Description

The SHOW VERSION command shows you what version of DEC/Test Manager you are using. The full text of this message should be included with Software Performance Reports (SPRs).

Example

```
Ⓢ DTM SHOW VERSION  
DEC/Test Manager Version V2.1
```

This example displays the version number of DEC/Test Manager.

DTM SPAWN

DTM SPAWN

Creates a subprocess of your current DEC/Test Manager session—your parent process.

Format **DTM SPAWN** *[command]*

Command Qualifiers

/[NO]CARRIAGE_CONTROL
/[NO]CLI[=cli]
/INPUT=file-spec
/[NO]KEYPAD
/[NO]LOGICAL_NAMES
/[NO]NOTIFY
/OUTPUT=file-spec
/PROCESS=subprocess-name
/[NO]PROMPT[=string]
/[NO]SYMBOLS
/[NO]WAIT

Defaults

See text.
See text.
See text.
/KEYPAD
/LOGICAL_NAMES
/NONOTIFY
See text.
See text.
See text.
/SYMBOLS
/WAIT

Command Qualifiers

/CARRIAGE_CONTROL
/NOCARRIAGE_CONTROL

Determines whether carriage control or line feed characters (or both) are prefixed to the DCL-prompt string of the subprocess. The default is the current setting of the parent process.

/CLI[=cli]
/NOCLI

Specifies an alternate command language interpreter (CLI) for the subprocess to use. The default is the CLI the parent process uses.

The CLI you specify must be located in SYS\$SYSTEM and have the file type EXE.

/INPUT=file-spec

Specifies an input file containing one or more commands for the spawned subprocess to execute. If you specify a command with an input file, the command is processed before the commands in the input file. The subprocess terminates when processing is complete. You cannot use wildcards in the file specification.

/KEYPAD
/NOKEYPAD

Determines whether DCL keypad symbols and the current DCL keypad state are copied from the parent process to the subprocess. The default is to copy any key definitions or states (or both) you have established with the DEFINE/KEY command. Use the /NOKEYPAD qualifier if you do not want the key settings to be copied.

/LOGICAL_NAMES
/NOLOGICAL_NAMES

Determines whether the system passes process logical names and logical name tables to the subprocess. The default is to copy all process logical names and logical name tables except those marked CONFINE or those created in executive or kernel mode.

/NOTIFY
/NONOTIFY

Determines whether a message is sent to your terminal to notify you that your subprocess has been completed or aborted. Do not specify /NOTIFY unless you also specify the /NOWAIT qualifier. The default is /NONOTIFY.

/OUTPUT=file-spec

Specifies the output file to which the output of the SPAWN operation is to be written. When you specify /NOWAIT, you should use /OUTPUT to specify an output other than SYS\$OUTPUT to prevent your terminal from being used by both processes simultaneously. The default is to direct output to the current SYS\$OUTPUT device.

/PROCESS=subprocess-name

Specifies the name of the subprocess to be created. The default name for the subprocess is USERNAME_#. The pound sign (#) denotes a unique number.

DTM SPAWN

/PROMPT[=string]

Specifies the prompt string for the subprocess. If you specify */PROMPT* but do not specify a string, the default prompt is displayed. The default is to copy the current prompt string from the parent process.

/SYMBOLS

/NOSYMBOLS

Determines whether the system passes DCL global and local symbols to the subprocess. The default is */SYMBOLS*.

/WAIT

/NOWAIT

Controls whether the system waits until the subprocess is completed before allowing more commands to be issued by the parent process. The */NOWAIT* qualifier allows you to enter more commands while the specified subprocess is running. When you specify */NOWAIT*, you should also specify */OUTPUT* to direct output to a file (rather than to the screen). This prevents your terminal from being used by both processes simultaneously. The default is */WAIT*.

Command Parameter

command

Specifies an optional command to be executed by the subprocess you are creating. If you specify the command parameter, you create a subprocess which executes the command and returns control to the DEC/Test Manager session when the command terminates. If you include the */INPUT* qualifier with the command parameter, the subprocess reads commands from the specified input file after the command executes. The command string cannot exceed 132 characters.

If you omit the command parameter, the SPAWN command creates a subprocess and attaches your terminal to it. You can return to your DEC/Test Manager session by logging out of the subprocess or by issuing the *ATTACH/PARENT* command. If you have created several subprocesses, you can switch between them using the *ATTACH/IDENTIFICATION* command.

Description

The SPAWN command creates a subprocess of your current DEC/Test Manager session—your parent process. The context of your DEC/Test Manager session is copied to the subprocess.

You can use the SPAWN command to leave DEC/Test Manager temporarily, to create another DEC/Test Manager session, or to edit a file and then return to your original DEC/Test Manager session.

Example

```
DTM> SPAWN MAIL
```

```
You have 1 new message.
```

```
MAIL>
```

This command spawns the MAIL utility from the DEC/Test Manager subsystem. Enter the ATTACH command to terminate the mail session and to return to the DEC/Test Manager subsystem.

DTM STOP

DTM STOP

Stops a collection that has been submitted to a batch queue.

Format **DTM STOP** *collection-name [remark]*

Command Qualifiers

/[NO]CONFIRM

/[NO]LOG

Defaults

/NOCONFIRM

/LOG

Command Qualifiers

/CONFIRM

/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm that you want to stop the collection.

The default is /NOCONFIRM.

/LOG

/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is /LOG.

Command Parameters

collection-name

Specifies the collection that is to be stopped. You cannot use wildcards to specify the collection-name parameter.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (""). If you do not provide a remark, you are prompted for one, but the collection is stopped even if you do not provide a remark.

Description

The STOP command stops the execution of a collection of tests that has been submitted to a batch queue. The batch job stops and is removed from the batch queue, necessary clean up is performed on the database, and you see a message. All tests that have completed are available for comparison and review.

Pressing CTRL/C stops a collection running interactively.

You must have the appropriate privileges to stop a collection submitted by someone else. If the person is in your UIC group, you must have the GROUP privilege. If the person is not in your UIC group, you must have the WORLD privilege.

Example

```
⌘ DTM STOP RNOCOLLECT /CONFIRM "Stopping run of RNOCOLLECT"  
Confirm stop of collection RNOCOLLECT [Y/N] (N): Y  
%DTM-S-COLSTOPPED, collection RNOCOLLECT has been stopped  
⌘
```

Stops collection RNOCOLLECT which is executing in batch.

DTM SUBMIT

DTM SUBMIT

Executes a collection of tests in batch.

Format **DTM SUBMIT** *collection-name [remark]*

Command Qualifiers

/[NO]CONFIRM
/[NO]KEEP
/[NO]LOG

Defaults

/CONFIRM
/KEEP
/LOG

Command Qualifiers

/CONFIRM
/NOCONFIRM

Controls whether DEC/Test Manager prompts you to confirm that you want to resubmit the collection if it has not been reviewed.

By default, you will not be prompted to confirm submission of a collection or resubmission of a collection that you have both executed and reviewed. If you have executed but not reviewed a collection, you will be prompted to confirm resubmission of the collection.

/NOKEEP
/KEEP

Controls whether the log file is deleted after it is printed.

The default is ***/KEEP***.

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is ***/LOG***.

Command Parameters

collection-name

Specifies the collection to execute in batch mode. You cannot use wild-cards to specify the collection-name parameter.

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (""). The collection is submitted even if you do not provide a remark.

Description

The SUBMIT command runs the batch command file created by the CREATE or RECREATE COLLECTION command for the tests in the specified collection. This command file runs the tests by executing the template file of each test description in the collection. It also runs any collection prologues or epilogues. Note that you cannot use the DCL SUBMIT command to execute a collection of tests in batch.

You can use the following DCL SUBMIT command qualifiers with the DTM SUBMIT command:

/AFTER	/[NO]CHARACTERISTICS	/CPUTIME
/[NO]HOLD	/[NO]LOG_FILE	/NAME
/[NO]NOTIFY	/[NO]PRINTER	/PRIORITY
/QUEUE	/[NO]USER	/WSDEFAULT
/WSEXTENT	/WSQUOTA	

You can find more information about these qualifiers in the *VAX/VMS DCL Dictionary*. Collections can be run more than once. When you rerun a collection of tests with the SUBMIT command, the collection is not recreated. The original collection is run again. For example, if you have changed any of the tests or the contents of any groups contained in the collection, these changes are not reflected in the resubmitted test run. If you issue the SUBMIT command to rerun a collection which you have not reviewed, DEC/Test Manager prompts you for confirmation before rerunning the collection.

Use the RECREATE command to recreate a collection with the most current version of all the files in the collection.

DTM SUBMIT

Example

```
⌘ DTM SUBMIT/NOTIFY/LOG_FILE=[]/QUEUE=SYS$LARGE MARGIN_COLLECT_1
%DTM-S-SUBMITTED, collection MARGIN_COLLECT_1 submitted
-DTM-I-TEXT, Job MARGIN_COLLECT_1 (queue SYS$LARGE entry 1000) started on SYS$LARGE
```

This example submits the collection MARGIN_COLLECT_1 to the batch queue SYS\$LARGE. The log file is left in the default directory and you are notified when the collection is finished running.

DTM VERIFY

Checks that the current DEC/Test Manager library is valid.

Format **DTM VERIFY** *[remark]***Command Qualifiers**

/[NO]LOG
/RECOVER

Defaults

/LOG
None.

Command Qualifiers

/LOG
/NOLOG

Controls whether DEC/Test Manager displays informational and success messages on your screen.

The default is */LOG*.

/RECOVER

Attempts to restore the current library or its collections to a usable state after a system crash or a process or job abort. It also cleans up the library by deleting files and directories that DEC/Test Manager does not own.

Command Parameter

remark

Specifies a string that contains a comment. This parameter is optional. If a remark includes any space characters, it must be enclosed in double quotation marks (""). If you do not provide a remark, you are prompted for one, but the library is verified even if you do not provide a remark.

DTM VERIFY

Description

The VERIFY command performs a series of checks on the current DEC/Test Manager library to confirm that the library structure and the library files are in a valid form.

If the library is valid, the command executes successfully. If the library is invalid, DEC/Test Manager informs you to use the /RECOVER qualifier to correct some of the errors the VERIFY command discovers.

If the VERIFY/RECOVER command encounters a subdirectory of the DEC/Test Manager library that contains files and the directory is not associated with a collection, DEC/Test Manager issues a confirmation message before deleting the directory. See Chapter 6 for more information on recovering an invalid DEC/Test Manager library.

Examples

- DTM> VERIFY
%DTM-I-VERFRE, free space list verified
%DTM-I-VERSTR, string list verified
%DTM-I-VERCOL, collection list verified
%DTM-I-VERGRO, group list verified
%DTM-I-VERTD, test description list verified
%DTM-I-VERVAR, variables list verified
%DTM-I-VERARC, archive list verified
%DTM-I-VERHEAD, user header information verified
%DTM-I-VERSPACE, contiguous space verified
%DTM-I-VERCOLDIR, collection directory structure verified
%DTM-S-VERIFIED, DEC/Test Manager library DRA1:[project.dtmlib] verified
DTM>

This command verifies that the current DEC/Test Manager library is valid.

- DTM> VERIFY/RECOVER
%DTM-S-RECNOTNEC, recovery is not necessary; DEC/Test Manager library
DRA1:[PROJECT.DTMLIB] is in a safe state
DTM>

This command verifies that the current DEC/Test Manager library is valid and informs you that you do not need to recover the library.

DEC/Test Manager Messages

This appendix lists the DEC/Test Manager messages alphabetically. The messages are accompanied by explanations and, where possible, suggested actions to recover from errors.

A.1 Message Display

DEC/Test Manager messages are displayed on the current output device. For an interactive user, this device is a terminal. If DEC/Test Manager is run in batch, messages are written into the log file.

A.1.1 Message Format

A DEC/Test Manager error message has the following format:

`%DTM-Severity_code-Message_name, Text of Message`

The fields are interpreted as follows:

- Severity_code is a single letter, one of the five described in the following section.
- Message_name is a name that uniquely identifies the message.
- Text of Message is a one- or two-sentence description of an event that has occurred.

A.1.2 Severity Codes

DEC/Test Manager messages range in purpose from confirming the successful completion of your last DEC/Test Manager command to notifying you of an error that caused the last command to be terminated.

The severity level of a message indicates the general nature of the message and is represented by one of the following codes:

- Success (S) messages indicate that DEC/Test Manager has performed your request.
- Informational (I) messages indicate certain kinds of information about the command you issued. For example, DEC/Test Manager informs you if it is waiting for the database to become available.
- Warning (W) messages indicate that DEC/Test Manager has encountered a minor conflict, but one that does not stop processing of your command.
- Error (E) messages indicate that DEC/Test Manager is unable to perform the requested command; you must correct the problem and reenter the command. Processing of the command might continue.
- Fatal (F) messages indicate that DEC/Test Manager is about to terminate because of a problem that prevents it from continuing any further. Processing of the command stops.

Some fatal problems can be resolved by issuing a VERIFY/RECOVER command. When appropriate, DEC/Test Manager directs you to issue this command. Chapter 6 describes the VERIFY/RECOVER command.

A.2 The DEC/Test Manager Messages

This section lists all DEC/Test Manager messages along with a brief explanation of each message and the recommended user action. If no user action is required, this section is omitted. The messages are listed in alphabetical order, by message name. A term enclosed in single quotation marks is variable information.

ABORTING, collection 'name' (entry 'number') is aborting in queue 'queue'

Explanation. The execution of the specified collection was stopped, and the job is in the process of aborting.

User Action. Wait for the job to finish aborting before trying to use the collection.

ABSTIM, 'qualifier' time value must be absolute

Explanation. Specify an absolute time value for the /BEFORE and /SINCE qualifiers.

User Action. Correct the time value and reenter the command.

ALLOC, could not allocate structure

Explanation. DEC/Test Manager could not allocate an internal structure.

User Action. Check quotas and SYSGEN parameters. See the secondary messages for more information.

ALPHACHAR, the first character in 'expression' must be alphanumeric

Explanation. You specified an expression that does not begin with a letter or a number.

User Action. Correct the expression and reenter the command.

ALRDYCOMPARED, collection 'name' has already been compared

Explanation. You tried to compare a collection that has already been compared.

ALRDYEXISTS, 'name' is already an 'object-name'

Explanation. You specified a name for a collection, group, test description, or variable that already exists in the current library.

User Action. Select another name and reenter the command.

ALRDYINGRP, object 'name' is already in group 'name'

Explanation. The test description or group already belongs to the specified group.

ALRDYUPD, the benchmark for result description 'name' has already been updated

Explanation. You tried to update a benchmark file that has already been updated.

ANSI_CRT, recording terminal was an ANSI crt, display terminal is not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

ASSIGNERR, unable to assign an I/O channel to device

Explanation. DEC/Test Manager could not access the specified device.

User Action. Check your quotas and SYSGEN parameters, make sure the device is available, and check the device's protection and ownership.

ASTERR, error declaring an AST

Explanation. A system service failed to create an AST.

User Action. Check the AST quotas.

ASTERROR, AST routine received the following error:

Explanation. An error occurred at AST level.

User Action. See the secondary messages for more information.

ASTRONLY, the only parameter recognized by this command is '*'

Explanation. You entered a parameter other than an asterisk (*). If you enter any parameter at all, it must be the asterisk wildcard character.

User Action. Reenter the command with no parameter or with an asterisk (*).

ATTACHERR, could not put pseudoterminal in ATTACH mode

Explanation. DEC/Test Manager could not put the PTY device into ATTACH mode.

User Action. Submit a Software Performance Report (SPR).

AVO_TERM, recording terminal had AVO option, display terminal does not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

BADCOLL, there is something wrong with collection 'collection-name'

Explanation. There is a problem with the specified collection.

User Action. Delete the collection or, if the collection is required, restore the library from a backup tape.

BADLENSTR, 'type' block length is 'length', should be 'length'

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BADLIB, there is something wrong with your DEC/Test Manager library

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BADORDSTR, 'type' block 'name' is out of order

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BADPTR, 'type' block has address 'address' outside range of database

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BADTYPSTR, 'type' block type is 'identifier', it should be 'identifier'

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BADVERSTR, 'type' block version is 'identifier', it should be 'identifier'

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BCKPTRSTR, 'type' back pointer is 'identifier' previous block is 'identifier'

Explanation. DEC/Test Manager has discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

BEGIN, your interactive test session is now beginning...

Explanation. Your recording session is beginning.

BMK_NOTSAVED, no benchmark file will be saved

Explanation. Your benchmark file will not be saved.

BMK_SAVED, benchmark has been saved in file 'name'

Explanation. Your benchmark file has been saved.

CANTRESUB, cannot resubmit this collection

Explanation. You cannot resubmit this collection.

CHECKCMS, check that SYS\$SHARE:CMSPROSHR.EXE is installed

Explanation. The DEC/CMS shareable image CMSSHR.EXE could not be activated.

User Action. Verify that the DEC/CMS start-up file was executed when your system booted and that SYS\$SHARE:CMSPROSHR.EXE is installed with the /OPEN/SHARE/PROTECTED qualifiers. Consult with your system manager.

CLREFERR, unable to clear a system event flag

Explanation. A system service failed to clear an event flag.

User Action. Check your quotas and SYSGEN parameters. Consult with your system manager.

CMDFILABORT, aborting execution of commands from command file 'name'

Explanation. An error occurred while DEC/Test Manager was executing commands from your start-up command file.

User Action. Correct the command that caused the error.

CMDNOTALLOWED, 'command' command is not allowed when Reviewing in Read_only mode

Explanation. You attempted to enter an INSERT or UPDATE command while reviewing a collection in read-only mode.

User Action. Exit the Review subsystem and then reenter it without the /READ_ONLY qualifier. As the primary reviewer, you can issue the INSERT and UPDATE commands.

CMDTOOLONG, command over 255 characters long, RECREATE
'name' will not work

Explanation. The original CREATE COLLECTION command was more than 255 characters long. Due to CLI restrictions, you cannot currently use the RECREATE command for this collection.

User Action. Use the DELETE COLLECTION command to manually delete the existing collection. Then reenter the original CREATE COLLECTION command.

CMP_ABORTED, test 'name' did not successfully complete comparison

Explanation. DEC/Test Manager could not compare the result file for the specified test with its benchmark file.

User Action. See the secondary message for more information.

CMPSTATIGNORED, comparison status qualifier ignored

Explanation. You specified both a result-description-name parameter and one or more comparison status qualifiers on a command. DEC/Test Manager ignored the comparison status qualifiers.

CNTSTR, block 'type' count is 'actual-count', it should be 'correct-count'

Explanation. The specified block type count is incorrect.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

CNVNOTNEC, conversion not necessary, library already version
'number'

Explanation. The old library you specified on the CONVERT LIBRARY command does not need to be converted.

COLLINUSE, collection 'name' is in use

Explanation. The collection you specified is currently in use.

User Action. Wait and reenter the command when the collection is not in use. If you see this message for a collection that you know is not in use, there is a problem with the collection. Use the VERIFY/RECOVER command to restore your library to a usable state.

COLNOTCOMP, collection 'name' has not been compared

Explanation. You tried to review a collection that has not yet been compared.

User Action. Compare the collection and reenter the command.

COLNOTRUN, collection 'name' has not been run

Explanation. You tried to compare or review a collection that has not yet been executed.

User Action. Execute the collection and reenter the command after the collection has finished executing.

COLNOTRVW, collection 'name' has not been reviewed

Explanation. You tried to resubmit or recreate a collection that has not been reviewed.

User Action. Review the collection and reenter the command.

COLSTATERR, collection is in an inconsistent state

Explanation. The collection is in an unusable state.

User Action. Use the VERIFY/RECOVER command to restore the collection to a usable state.

COLSTOPPED, collection 'name' has been stopped

Explanation. DEC/Test Manager stopped execution of the specified collection.

COMPARED, collection 'name' compared

Explanation. DEC/Test Manager compared the specified collection.

CONCLUDED, your interactive test session has concluded

Explanation. You terminated the recording session.

CONTROL C, operation aborted by CTRL/C

Explanation. You pressed CTRL/C to abort an operation in progress.

User Action. Reenter the command to restart the operation.

CONVERTED, version 'number' library successfully converted to version 'number'

Explanation. DEC/Test Manager converted the library.

COPIED, test description 'name' copied

Explanation. DEC/Test Manager copied the specified test description.

CREATED, 'object' 'name' created

Explanation. DEC/Test Manager created the specified collection, group, test description, variable, or library.

CR_FILLS, recording terminal needs no crfill, display terminal does

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

CTRLCERR, error enabling CTRL/C ASTs

Explanation. DEC/Test Manager could not enable CTRL/Cs on your terminal.

User Action. Check your terminal's protection, existence, and ownership.

CURRCOMPARE, collection 'name' is currently being compared

Explanation. You tried to compare a collection that is currently being compared.

User Action. Wait and reenter the command after the collection has been compared.

CURRTERM, characteristics of current terminal will be used

Explanation. DEC/Test Manager could not extract the terminal characteristics information from the specified SESSION file because it could not find the file. DEC/Test Manager will use the terminal characteristics of your current terminal.

DASSGNERR, unable to deassign the I/O channel to 'device'

Explanation. A system service could not release control of the specified device.

User Action. Check the availability, protection, and ownership of the device.

DCTRL, error disabling control characters

Explanation. A system service failed to disable CTRL/Y handling.

User Action. See the secondary messages for more information.

DEALLOC, failed to deallocate 'structure'

Explanation. An internal structure could not be deallocated.

User Action. See the secondary messages for more information.

DEC_CRT1, recording terminal was a DEC crt, display terminal is not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

DEC_CRT2, recording terminal was a DEC crt, display terminal is not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

DEFAULTDIR, default directory cannot be a DEC/Test Manager library or any of its subdirectories

Explanation. You cannot set your default directory to the DEC/Test Manager library or to any of its subdirectories.

User Action. Change your default directory to be different from the DEC/Test Manager library or any of its subdirectories, and reenter the command.

DEFAULTED, 'type' file-name defaulted to 'name'

Explanation. DEC/Test Manager provided a default file-name or extension.

User Action. Check the default name or extension to verify that it agrees with the existing file-name or extension.

DEFCANCEL, default 'item' canceled

Explanation. DEC/Test Manager canceled your current default benchmark directory, template directory, or collection prologue or epilogue file.

DELETED, 'object' 'name' deleted

Explanation. DEC/Test Manager deleted the specified collection, group, test description, or variable.

DELETIONS, 'count' deletion(s) completed

Explanation. DEC/Test Manager deleted the indicated number of items.

DELSINRUN, test description 'name' has been deleted since this collection was run

Explanation. DEC/Test Manager cannot insert this test description into the group created from the Review subsystem because the test description has been deleted.

DEVUNKNOWN, device type unknown—VT100 assumed

Explanation. Your terminal is of an unknown device type. DEC/Test Manager will record the interactive terminal session as if you were using a VT100 series terminal.

DIFFERENT, files for test 'name' are different

Explanation. The COMPARE command detected differences between the result and benchmark files for the specified test description.

DIREXISTS, directory 'name' already exists

Explanation. DEC/Test Manager could not create the specified collection. The collection subdirectory it tried to create already exists.

DISPLAYSUP, display will be suppressed since device 'name' is not a terminal

Explanation. You entered a PLAY or RUN command with SYS\$OUTPUT specifying a device other than a terminal. The test will execute properly, but you will not be able to monitor its progress visually.

ECTRL, error enabling control characters

Explanation. A system service failed to enable CTRL/Y handling.

User Action. See the secondary messages for more information.

EIGHTBIT, recording terminal handles 8-bit, display terminal does not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

EMPTYGROUP, group 'name' contains no test descriptions

Explanation. The specified group contains no test descriptions.

User Action. Reenter the command without the named group, or examine the library to see if this group should contain test descriptions.

ENDPTRSTR, 'type' end pointer is 'identifier', last block is 'identifier'

Explanation. The last block type is incorrect.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

ERRCOMP, error comparing test 'name'

Explanation. DEC/Test Manager could not compare the result and benchmark files for the specified test.

User Action. See the secondary message for more information.

ERRCREASSOC, error associating variable 'name' with test description 'name'

Explanation. DEC/Test Manager could not associate the specified variable and test description.

User Action. See the secondary message for more information.

ERRDELETIONS, 'count' deletion(s) completed with 'count' errors

Explanation. DEC/Test Manager encountered one or more errors while deleting items.

ERRDELFIL, error deleting file 'name'

Explanation. DEC/Test Manager could not delete the specified file.

User Action. Check the file protection and reenter the command.

ERREMOVALS, 'count' removal(s) completed with 'count' error(s)

Explanation. DEC/Test Manager encountered one or more errors while removing test descriptions or groups.

ERRINCOLL, error in collection 'name'

Explanation. The specified collection contains errors.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

ERRINSERTIONS, 'count' insertion(s) completed with 'count' error(s)

Explanation. DEC/Test Manager encountered one or more errors while inserting test descriptions or groups.

ERRINTD, error in test description 'name'

Explanation. The specified test description contains errors.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

ERRMODIFIES, 'count' modification(s) completed with 'count' error(s)

Explanation. DEC/Test Manager encountered one or more errors while modifying test descriptions or variables.

ERROVERRIDE, error associating new value of variable 'name' with test description 'name'

Explanation. DEC/Test Manager encountered one or more errors while associating an override value with the specified variable.

User Action. See the secondary message for more information.

ERRPAREXP, error parsing 'type' expression

Explanation. You entered a collection, group, test description, or variable expression with illegal syntax.

User Action. Correct the expression and reenter the command. See the secondary message for more information.

ERRSUBDIR, library subdirectory 'name' matches no existing collection

Explanation. A subdirectory exists in the DEC/Test Manager library that is not related to a valid collection.

User Action. Delete the subdirectory from the DEC/Test Manager library.

ERRUPDATES, 'count' update(s) completed with 'count' error(s)

Explanation. DEC/Test Manager encountered one or more errors while updating benchmark files.

ERRVERARC, archive list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERCOL, collection list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERCOLFIL, one or more files missing from collection sub directory

Explanation. DEC/Test Manager discovered errors in the collection directory structure.

User Action. Recreate the collection.

ERRVERFRE, free space list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERGRO, group list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERHEAD, user header information verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERSPACE, contiguous space verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERSTR, string list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERTD, test description list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

ERRVERVAR, variables list verified with errors

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

EXIT, leaving Review subsystem

Explanation. DEC/Test Manager exited the Review subsystem.

EXITERR, error on exit from Review subsystem

Explanation. DEC/Test Manager encountered one or more errors while exiting the Review subsystem.

EXPRIGNORED, 'expression type' expression ignored

Explanation. DEC/Test Manager ignored the specified expression because the command is syntactically incorrect. DEC/Test Manager will execute the command as if you had not entered the expression.

User Action. See the secondary message for more information.

EXTRACTED, input file "file-name" created

Explanation. DEC/Test Manager extracted the INPUT file from the specified SESSION file.

FILENOTEXIST, 'type' file 'name' does not exist

Explanation. The specified file does not exist. When a file (except the benchmark file) is named in a test description, the file must exist when a collection containing the test description is created.

User Action. Refer to Chapter 3 to determine whether the missing file is one that you can create and modify.

FILMOVED, 'template/benchmark' file 'name' moved to CMS library

Explanation. 'library-name' DEC/Test Manager moved the specified benchmark or template file from your default directory to the specified DEC/CMS library.

FILNAMERR, error in 'type' file-name 'name'

Explanation. There is a syntax error in the specified file-name.

User Action. Reenter the command with a proper file-name. See the secondary message for more information.

FILNOTMOVED, 'template/benchmark' file 'name' not moved to CMS library 'library name'

Explanation. DEC/Test Manager was unable to move the specified file into the specified DEC/CMS library.

User Action. See the subsequent DEC/CMS messages to determine the problem. Then use DEC/CMS directly to place the file into the DEC/CMS library.

FREEEFERR, unable to deallocate a system event flag

Explanation. A system service failed to deallocate an event flag.

User Action. Check quota and SYSGEN parameters. Consult with your system manager.

FROMSELF, cannot remove group 'name' from itself

Explanation. You tried to remove the specified group from itself. A group cannot be a member of itself, so you cannot remove a group from itself.

GETDVIERR, could not get information about device 'dev'

Explanation. A system service failed to obtain information about the specified device.

User Action. Verify that the device exists. Check the device protection and ownership.

GETEFERR, unable to allocate a system event flag

Explanation. A system service failed to allocate an event flag.

User Action. Check your quotas and SYSGEN parameters. Consult with your system manager.

HASFILES, directory 'name' contains files

Explanation. You tried to create a DEC/Test Manager library in a directory that is not empty.

User Action. Delete the files and reenter the CREATE LIBRARY command for the same directory or create the library in a different directory.

HASMEMBERS, group 'name' contains one or more groups or test descriptions

Explanation. You tried to delete a group that is not empty.

User Action. Remove any groups and test descriptions from the group, then reenter the command.

HASREFERENCE, variable 'name' referenced by a test description

Explanation. You tried to delete a variable that is referenced by one or more test descriptions.

User Action. Enter the SHOW VARIABLE/TEST_DESCRIPTION command to list the tests that reference this variable. Then remove the references to the variable with the MODIFY TEST_DESCRIPTION command and reenter the DELETE VARIABLE command.

HISNOTSTM, history file record format is not stream_lf

Explanation. The library history file (00DTM.HIS) must be in stream_lf (line feed) format. Note that if you edit the history file, the format may no longer be stream_lf.

User Action. Use the VERIFY/RECOVER command to correct the file format.

HISTDEL, 'count' history records deleted

Explanation. DEC/Test Manager deleted some or all of your library history. By default, the deleted history information is placed in the file HISTORY.DMP in your default directory.

HOLDING, collection 'name' (entry 'number') is holding in queue 'queue'

Explanation. The specified collection is waiting to execute.

IDENTICAL, files for test 'name' are identical

Explanation. The result and benchmark files for this result description are identical.

ILLCHAR, illegal character in 'name'

Explanation. 'name' contains a character that is not allowed in this context.

User Action. Correct the expression and reenter the command.

ILLEGALDEV, illegal device name specified

Explanation. You included an illegal device specification in a library directory specification or in a file specification.

User Action. Correct the device specification and reenter the command.

ILLFILEINLIB, 'type' file 'name' contains illegal DEC/Test Manager library specification

Explanation. The file specification cannot refer to the DEC/Test Manager library.

User Action. Correct the file specification and reenter the command.

ILLPRIORITY, /PRIORITY not in range 0 to 255

Explanation. You tried to submit a collection to the batch queue with a priority value outside the allowed range.

User Action. Correct the priority value and reenter the command.

ILLQUAL, 'name' is an illegal qualifier; use /TEST_DESCRIPTION or /GROUP

Explanation. You cannot use the specified qualifier in a test-group-expression. Only the parameter qualifiers, /GROUP and /TEST_DESCRIPTION, can be used here.

User Action. Verify the validity and position of the qualifier, then reenter the command.

ILLRECORD, unrecognized record type 'type'

Explanation. A record in the SESSION file being processed begins with the specified unrecognized type designator.

User Action. The illegal record should be deleted or modified, or a previous correct version of the file should be used.

ILLTIME, /CPUTIME value is not a delta time

Explanation. You tried to submit a collection with a /CPUTIME qualifier value that is not a delta time.

User Action. Correct the value and reenter the command.

ILLWSDEFAULT, /WSDEFAULT not in range 1 to 65535

Explanation. You tried to submit a collection with a /WSDEFAULT qualifier value outside the allowed range.

User Action. Correct the value and reenter the command.

ILLWSEXTENT, /WSEXTENT not in range 1 to 65535

Explanation. You tried to submit a collection with a /WSEXTENT qualifier value outside the allowed range.

User Action. Correct the value and reenter the command.

ILLWSQUOTA, /WSQUOTA not in range 1 to 65535

Explanation. You tried to submit a collection with a /WSQUOTA qualifier value outside the allowed range.

User Action. Correct the value and reenter the command.

INPERR, command line input error

Explanation. You entered a command containing a syntax error.

User Action. See the secondary messages for more information.

INSERTED, 'object' 'name' inserted into group 'name'

Explanation. DEC/Test Manager inserted the group or test description into the specified group.

INSERTIONS, 'count' insertions completed

Explanation. DEC/Test Manager completed the specified number of insertions.

INTOSELF, cannot insert group 'name' into itself

Explanation. You tried to insert the specified group into itself. A group cannot be a member of itself.

INUSE, DEC/Test Manager library 'directory-spec' is in use, please wait

Explanation. The DEC/Test Manager library is currently being used by someone else. DEC/Test Manager will wait and automatically continue execution of your command as soon as the library is free.

INVTRMCHR, invalid termination character 'string' specified

Explanation. The string you specified for the /TERMINATION_CHARACTER qualifier is invalid.

User Action. Correct the string specification and reenter the command.

IOERROR, error 'opening' file 'name'

Explanation. An I/O error occurred while DEC/Test Manager was either reading, writing, opening, or closing the specified file. The type of file, the specified action, and the file-name are included in the message text.

User Action. Verify that the file exists and check the file protection and ownership. See the secondary messages for more information.

ISMEMBER, 'object' 'name' is a member of group 'name'

Explanation. You tried to delete a group or test description that belongs to another group.

User Action. Remove the test description or group from all groups to which it belongs, then reenter the command.

LF_FILLS, recording terminal needs no lfill, display terminal does

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

LIBIS, DEC/Test Manager library is 'directory-spec'

Explanation. Your DEC/Test Manager library is now defined to be the specified directory.

LOWERCAS, recording terminal had lowercase, display terminal does not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

MECHFORM, recording terminal supports FF, but display terminal does not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

MECH_TAB, recording terminal supported tabs, display terminal does not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

MISBLKSTR, a 'type' block was not hit during pass 1

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

MODIFICATIONS, 'count' modification(s) completed

Explanation. DEC/Test Manager modified the specified number of test descriptions or variables.

MODIFIED, 'object' 'name' modified

Explanation. DEC/Test Manager modified the specified test description or variable.

MRKFORINSERT, test_description 'name' marked for insertion

Explanation. DEC/Test Manager marked the specified test description for insertion into the group created when you exited the Review subsystem.

MSSBLKSTR, there were 'count' 'identifier' type blocks found on pass 1,
there were 'count' blocks found on pass 2

Explanation. DEC/Test Manager discovered an inconsistency or error in your DEC/Test Manager library that it cannot correct.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

MUSTBEDIR, 'string' must be a directory specification

Explanation. DEC/Test Manager expected a directory specification where you entered the specified string of characters.

User Action. Correct the parameter and reenter the command.

MUSTBEGLOB, variable 'name' must be global for CREATE
COLLECTION to override

Explanation. The variable you specified on the CREATE COLLECTION command is a local variable.

User Action. Verify that you entered the correct variable-name, then reenter the command with a variable that is global in scope.

NETNOTALL, network access not allowed

Explanation. DEC/Test Manager does not allow you to access files over a network.

User Action. Remove all network references and reenter the command.

NEWASSOCVAL, superseding old association value between variable
'name' and test description 'name'

Explanation. This variable and test description are already associated. The new value you specified supersedes the existing value.

NEWDEF, 'file' is the new default collection 'prologue/epilogue'

Explanation. You specified a new default collection prologue or epilogue file.

NEWWIDTH, recording terminal has width n, display terminal has width m

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

NEW_TERM, recording terminal was a 'terminal-type', display terminal is a 'terminal-type'

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

NEWTEST, test 'name' is a new test

Explanation. No benchmark file for the test existed, so the test is new.

NOASSOC, variable 'name' was not associated with test-description 'name'

Explanation. You attempted to remove a nonexistent association between the specified variable and test description.

NOATTACH, could not attach to process

Explanation. DEC/Test Manager could not issue the specified ATTACH command.

User Action. Verify the existence of the process to which you want to attach and verify that it is part of your job. See the secondary messages for more information.

NOCOMPARE, collection 'name' not compared

Explanation. DEC/Test Manager could not compare the specified collection.

NOCONVERT, error converting your 'V1' library to 'V2'

Explanation. DEC/Test Manager could not convert your library.

User Action. See the secondary message for more information.

NOCOPY, error copying 'object' 'name'

Explanation. DEC/Test Manager could not copy the specified collection, group, test description, or variable.

User Action. See the secondary message for more information.

NOCREATE, error creating 'object' 'name'

Explanation. DEC/Test Manager could not create the specified collection, group, test description, variable, or library.

NOCURRES, you are not currently positioned at a result description

Explanation. You entered a Review subsystem command which requires that you be positioned at a result description.

User Action. Enter a Review subsystem command to select a result description, then reenter the command.

NODEFINE, your DEFINE/KEY command is syntactically incorrect

Explanation. There is an error in the DEFINE/KEY command you entered.

User Action. See the secondary messages for more information.

NODEL, cannot delete 'object'

Explanation. DEC/Test Manager could not delete the specified collection, group, test description, or variable.

User Action. See the secondary message for more information.

NODELETE, error deleting 'object' 'name'

Explanation. DEC/Test Manager could not delete the specified collection, group, test description, or variable.

User Action. See the secondary message for more information.

NODELETIONS, no 'name' deletions performed

Explanation. DEC/Test Manager could not delete the specified items.

User Action. See the secondary message for more information.

NODELFUTURE, cannot delete history of future events

Explanation. You specified a time that is later than the current time.

User Action. Correct the time value and reenter the command.

NODISPLAY, could not display benchmark for test 'name'

Explanation. The DISPLAY/BENCHMARK command could not display the specified benchmark file.

User Action. See the secondary messages for more information.

NOEXTRACT, input file not extracted from "session file"

Explanation. DEC/Test Manager could not extract an INPUT file from the specified SESSION file.

User Action. See the secondary message for more information.

NOHIS, no history records found

Explanation. The current DEC/Test Manager library contains no history records for the specified object.

NOINSERT, error inserting 'object' 'name' into group 'name'

Explanation. DEC/Test Manager could not insert the test description or group into the specified group.

User Action. See the secondary message for more information.

NOINSERTIONS, no 'type' insertions performed

Explanation. DEC/Test Manager did not perform the specified insertions.

User Action. See the secondary message for more information.

NOMARK, cannot mark result description 'name' for insertion

Explanation. DEC/Test Manager could not mark the named result description for insertion on Exit from Review.

User Action. See secondary message for more information.

NOMATCH, no match was found for 'type' expression 'expression'

Explanation. DEC/Test Manager could not find any item in the library or collection to match your command line expression.

User Action. Verify the spelling and meaning of your expression, then reenter the command.

NOMODARG, arguments do not specify any modifications to 'object'

Explanation. DEC/Test Manager did not modify the test description or variable because you did not specify any fields to be modified.

User Action. Reenter the command with one or more qualifiers specifying fields to modify.

NOMODIFIES, no 'type' modifications performed

Explanation. DEC/Test Manager performed no modifications.

User Action. See the secondary message for more information.

NOMODIFY, error modifying 'object' 'name'

Explanation. DEC/Test Manager could not modify the specified test description or variable.

User Action. See the secondary message for more information.

NOMORE, no more 'type' result descriptions found

Explanation. DEC/Test Manager could not find a result description matching the result-description-expression.

NOMOVE, result description position not changed

Explanation. DEC/Test Manager did not move you to the specified result description because the command you entered was syntactically incorrect.

User Action. See the secondary message for more information.

NONTTYRECORDER, recording may only be performed when running on a terminal

Explanation. You cannot record a terminal session while executing in batch.

User Action. Verify that SYS\$INPUT and SYS\$OUTPUT are specified as terminals.

NOPARTWILD, no partial wildcards allowed in result description expression

Explanation. Partial wildcards are not allowed as parameters to the SELECT Review subsystem command.

User Action. Reenter the command specifying either a result-description-name or the asterisk (*) full field wildcard, or no parameter.

NOPRINT, cannot print 'file type' 'name'

Explanation. DEC/Test Manager could not print the specified file.

User Action. See the secondary message for more information.

NOPRINTQD, error submitting job to print queue

Explanation. DEC/Test Manager could not submit the specified file(s) to the print queue.

User Action. See the secondary message for more information.

NORECOVER, error recovering library

Explanation. DEC/Test Manager could not recover your library.

User Action. Restore your library from a backup tape.

NORECREATE, error recreating collection 'name'

Explanation. DEC/Test Manager could not recreate your collection.

NOREF, unable to reference 'directory'

Explanation. You tried to execute a DEC/Test Manager command without first selecting a DEC/Test Manager library.

User Action. Use the SET LIBRARY command to select a DEC/Test Manager library, then reenter the command.

NOREMARK, error adding remark to DTM library

Explanation. DEC/Test Manager could not enter your remark in the library history file.

User Action. See the secondary messages for more information.

NOREMOVAL, error removing 'object' 'name' from group 'name'

Explanation. DEC/Test Manager could not remove the group or test description from the specified group.

User Action. See the secondary message for more information.

NOREMOVALS, no 'type' removals performed

Explanation. DEC/Test Manager performed no removals.

NORESUBMIT, collection 'name' cannot be resubmitted

Explanation. DEC/Test Manager could not resubmit the specified collection.

User Action. See the secondary message for more information.

NORETRIEVE, could not retrieve 'information' for test 'name'.

Explanation. DEC/Test Manager could not access the specified information for the specified test description.

NOREVIEW, error reviewing collection

Explanation. DEC/Test Manager could not review the specified collection.

User Action. See the secondary message for more information.

NOSET, could not set default 'item'

Explanation. DEC/Test Manager could not establish the specified default collection prologue or epilogue file, or the specified default library benchmark or template directory.

User Action. Correct the expression and reenter command.

NOSHOW, error showing 'object' 'name'

Explanation. DEC/Test Manager could not display the specified result, benchmark, or differences file; or it could not display the test description, variable, group, or collection.

NOSINCE, error executing /SINCE operation

Explanation. DEC/Test Manager could not display the library history as specified by the /SINCE qualifier.

User Action. See the secondary messages for more information.

NOSPAWN, could not spawn a subprocess

Explanation. DEC/Test Manager could not spawn a subprocess.

User Action. Verify your process quotas and job limits, then reenter the command.

NOSPEC, no 'item' specified

Explanation. No default benchmark or template directory, or default collection prologue or epilogue file exists.

NOSRCHLST, search lists are not allowed in this context: 'name'

Explanation. You cannot use a search list here.

NOSTOP, collection 'name' was not stopped

Explanation. DEC/Test Manager did not stop the collection.

NOSUBMIT, collection 'name' cannot be submitted

Explanation. DEC/Test Manager could not submit the specified collection.

User Action. See the secondary message for more information.

NOSUCHJOB, job 'number' does not exist in queue 'name'

Explanation. DEC/Test Manager could not find an executing batch job for the collection.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state, then reenter the command.

NOTAPPLIC, the /STRING and /NUMERIC qualifiers are not applicable to logical variable 'name'

Explanation. DEC/Test Manager did not modify the variable type since the variable has the logical usage and cannot also have the string or numeric type.

User Action. Verify that the current usage for the variable is correct.

NOTCHANGED, the default 'name' directory was not changed

Explanation. DEC/Test Manager could not change the specified directory.

NOTCRELIB, first history record is not DTM CREATE LIBRARY transaction

Explanation. The first record of every history file should be a DTM CREATE LIBRARY transaction. It is likely that the history file has been edited.

NOTDTMLIB, 'directory_spec' is not a valid DEC/Test Manager library

Explanation. You specified a directory that is not a valid DEC/Test Manager library.

User Action. Correct the directory specification and reenter the command.

NOTESTCMP, no tests in collection 'name' could be compared

Explanation. No tests ran in this collection; consequently, no result files were produced.

User Action. This collection can be resubmitted or recreated.

NOTFINISHED, collection 'name' has not finished running

Explanation. You tried to compare a collection that is still executing.

User Action. Reenter the command when the collection has finished executing.

NOTFOUND, 'object' not found

Explanation. DEC/Test Manager could not find the specified collection, group, test description, or variable.

NOTINTER, test description 'name' is not interactive

Explanation. You performed an operation reserved for interactive tests on a noninteractive test.

NOTMEMBER, 'type' 'name' is not currently a member of group 'name'

Explanation. You selected a group or test description for removal from a specified group, but the test description or group is not currently a member of the specified group.

NOTTRANSLATE, unrecognized sequence "sequence" will not be translated

Explanation. DEC/Test Manager cannot provide a translation for a special string or for a recording function or nonprinting text while creating the INPUT or SESSION file. DEC/Test Manager will copy the untranslated sequence to the file.

User Action. Verify that the sequence is correct. If it is correct, use a text editor to include the correct translation in the file being created.

NOTRUN, collection 'name' was not found running in the batch queues

Explanation. You tried to stop a collection that is not currently executing in any batch queue.

NOTYETRUN, collection 'name' has not yet been run

Explanation. You tried to review or compare a collection that has not yet been executed.

User Action. Use the DTM RUN or SUBMIT command to execute the collection, then reenter the command.

NOUPDATE, the benchmark file for result description 'name' has not been updated

Explanation. DEC/Test Manager encountered an error while updating the benchmark file.

User Action. See the secondary message for more information.

NOVERIFY, DEC/Test Manager library 'name' not verified

Explanation. DEC/Test Manager encountered errors that it cannot correct while trying to verify the library.

User Action. Restore the library from a backup tape to ensure that you are using a consistent library.

NOWLDCARD, wildcards are not allowed in this context: 'string'

Explanation. You specified a wildcard character where none is allowed.

User Action. Replace the expression with a name and reenter the command.

NO_PC_DEV, your system does not contain the PC: device

Explanation. The device PC0: does not exist on your system.

User Action. Verify that the PCDRIVER is installed on your system and check the system start-up procedure to ensure that the PCDRIVER is being loaded.

NULLEXPR, a null 'type' expression is not allowed in this context

Explanation. You entered an empty string for a required parameter.

User Action. Reenter the command with a valid parameter.

NULLNAME, a null file-name for qualifier 'name' is not permitted in this context

Explanation. You entered an empty string for a required qualifier.

User Action. Reenter the command with a valid qualifier.

NULLNUM, numeric symbol variables cannot have a null value

Explanation. Numeric symbols with null values generate invalid DCL symbol assignment statements of the form VARIABLE =. This is not allowed.

User Action. If your MODIFY or CREATE command specified both the /VALUE=" " and the /NUMERIC qualifiers, reenter the command with one qualifier or the other but not both. If you were attempting to modify an existing variable, examine it with the SHOW VARIABLE command to verify that your modification is sensible.

NULLOG, logical variables cannot have a null value

Explanation. Logicals with null values generate invalid DCL DEFINE statements.

User Action. If your MODIFY or CREATE command specified both the /VALUE=" " and the /LOGICAL qualifiers, reenter the command with one qualifier or the other but not both. If you were attempting to modify an existing variable, examine it with the SHOW VARIABLE command to verify that your modification is sensible.

OLDLIB, your current library is an old version, please use the CONVERT command

Explanation. Your current library was created by a previous (incompatible) version of DEC/Test Manager.

User Action. See Chapter 2 for instructions for converting your library.

OPENIN, error opening 'name' as input

Explanation. The COMPARE command could not open the specified file for input.

OPENOUT, error opening 'name' for output

Explanation. The COMPARE command could not open the specified file for output.

ORIGBENMISS, original benchmark 'name' no longer exists; current benchmark is 'name'

Explanation. The original benchmark file has been changed.

OVERRUN, data overrun using PC: device

Explanation. Your program output too much data at a rate that DEC/Test Manager could not handle.

User Action. Verify that your parent process is not running at a low priority, or rerun your program when the system is not as heavily loaded.

PAGESIZE, recording terminal has PAGESIZE n, display terminal PAGESIZE m

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

PARTCMP, collection 'name' was partially compared

Explanation. DEC/Test Manager partially compared the specified collection.

PC_CHECK, check that PCDRIVER is installed and PC0: and PTY0: exist

Explanation. PCDRIVER is not present or is improperly installed.

User Action. Check the system start-up procedure to ensure that PCDRIVER is loaded. Verify that the devices PC0:, PTY0:, and VTA0: exist.

PC_READ, error reading from PC: device

Explanation. An error occurred while reading from the PC: device.

User Action. Check quotas and protections.

PC_WRITE, error writing to PC: device

Explanation. An error occurred while writing to the PC: device.

User Action. Check default protections and process quotas.

PENDING, collection 'name' (entry 'number') is pending in queue 'queue'

Explanation. The collection is waiting for room in the batch queue before beginning its execution.

PLAYFAILED, play of session 'name' has failed

Explanation. DEC/Test Manager could not execute the specified test.

User Action. See the secondary messages for more information.

PRINT, file 'name' of test 'name' selected for printing

Explanation. DEC/Test Manager marked the specified file for printing.

PRINTQD, print job has been sent to the print queue

Explanation. DEC/Test Manager submitted the specified job to the print queue.

PROCEEDING, proceeding with command execution

Explanation. Your library is now free and DEC/Test Manager is continuing with execution of your command.

PTY_ERR, could not create pseudoterminal for 'name'

Explanation. DEC/Test Manager could not create a PTY: device like the specified device.

User Action. Check the SYSGEN parameters and process quotas, then verify that PCDRIVER is loaded. See the secondary messages for more information.

QUALEXPRCONFLICT, the '/'qualifier' conflicts with the 'expression type' expression

Explanation. The specified qualifier and expression cannot both be included on the same command.

User Action. See the primary message for more information.

REASSURE, nonetheless, result file will be properly built

Explanation. Previous informational messages stated that the terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded. This message assures you that DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file.

RECGRO, inserting group 'name' into group 'name' would create a recursive group

Explanation. You cannot build a recursive group, a group that contains itself.

User Action. Create a new group with a different name and insert the group into that group.

RECNOTNEC, recovery is not necessary; DEC/Test Manager library 'directory-spec' is in a safe state

Explanation. The DEC/Test Manager library is not in need of recovery. DEC/Test Manager made no changes to the library.

RECORDED, test 'name' has been successfully recorded

Explanation. DEC/Test Manager recorded the terminal session.

RECORDFAILED, session file for test description 'name' has not been recorded

Explanation. DEC/Test Manager could not record the terminal session.

User Action. See the secondary messages for more information.

RECOVERED, DEC/Test Manager library 'directory-spec' recovered

Explanation. Your DEC/Test Manager library has been recovered and is ready for use.

RECREATED, collection 'name' has been recreated

Explanation. DEC/Test Manager deleted and recreated the specified collection.

REMARK, remark added to history file

Explanation. DEC/Test Manager added your remark to the history file.

REMOVALS, 'count' removals completed

Explanation. DEC/Test Manager performed the specified number of removals.

REMOVED, 'object' 'name' removed from group 'name'

Explanation. DEC/Test Manager removed the group or test description from the specified group.

RESDESCERR, the information stored in this result description is in error

Explanation. One of the files associated with the result description is in error or has been deleted.

User Action. You cannot review this result description as it is. See Chapter 5 for information on using the REVIEW INSERT command and on reexecuting the test.

RESERVNAM, 'name' is reserved for DEC/Test Manager use only

Explanation. The name you specified is reserved for use only by DEC/Test Manager.

User Action. Reenter the command with a different name.

RESTRMERR, could not restore terminal characteristics for 'terminal'

Explanation. A system service failed to restore your terminal's characteristics.

User Action. See the secondary messages for more information.

RESUBMITTED, collection 'name' has been resubmitted

Explanation. The collection is now executing in batch.

RUNFAILED, run of collection 'name' has failed

Explanation. DEC/Test Manager could not execute the specified collection.

User Action. See the secondary messages for more information.

RUNNING, collection 'name' (entry 'number') is running in queue 'name'

Explanation. The collection is executing in the specified queue.

SCOPETRM, recording terminal was a scope, but display terminal is not

Explanation. The terminal on which you are monitoring an executing test has different characteristics than the terminal on which the test was recorded.

User Action. DEC/Test Manager is executing the test correctly (even though the display may look incorrect) and will create a proper result file. Use the Review subsystem SHOW/RESULT command to display the result file.

SENSEMODEERR, could not get terminal characteristics for 'terminal'

Explanation. A system service failed to obtain the specified terminal's device characteristics.

User Action. Verify that the terminal exists and check the terminal's protection and ownership.

SESSION_BAD, session file 'name' has invalid format

Explanation. The SESSION file contains invalid data. Most likely, you edited a SESSION file created by DEC/Test Manager or you created a SESSION file without using the appropriate DEC/Test Manager commands.

User Action. If you modified the SESSION file, verify your changes, or replace the SESSION file with a backup version. Make sure that the first record in the SESSION file is a 12-byte terminal characteristics block. See Appendix B for more information.

SESSION_READ, error reading session file 'name'

Explanation. DEC/Test Manager could not read from the specified SESSION file.

User Action. Verify that the file exists and check the file's protection and ownership. See the secondary messages for more information.

SESSION_WRITE, error writing session file 'name'

Explanation. DEC/Test Manager could not write data to the specified file.

User Action. Verify that the file exists and check the file's protection and ownership. See the secondary messages for more information.

SETMODEERR, could not change terminal characteristics for 'terminal'

Explanation. A system service failed to change the specified terminal's device characteristics.

User Action. Verify that the terminal exists and check the terminal's protection and ownership.

SETTRMERR, could not change terminal characteristics for 'terminal'

Explanation. A system service failed to change your terminal's characteristics.

User Action. Verify the terminal's protection and ownership.

SIMTERM_CLS, could not deallocate simulated terminal

Explanation. An error occurred while terminating terminal simulation.

User Action. See the secondary messages for more information.

SIMTERM_ERR, error simulating terminal

Explanation. An error occurred while simulating your terminal.

User Action. See the secondary messages for more information.

SIMTERM_OPN, could not create simulated terminal

Explanation. DEC/Test Manager could not simulate your terminal.

User Action. See the secondary messages for more information.

STARTDELHIS, no deletable history records before 'date'

Explanation. You specified a date for the /BEFORE qualifier prior to the date of the first history record that can be deleted. The CREATE LIBRARY record cannot be deleted.

User Action. Correct the date and reenter the command.

STARTHIS, library history starts at 'date'

Explanation. You specified a date for the /BEFORE qualifier prior to the date the library was created.

User Action. Correct the date and reenter the command.

STRTOOLONG, string of 'count' characters is too long

Explanation. The name you specified is too long.

User Action. Reenter the command with a shorter name.

SUBMITTED, collection 'name' submitted

Explanation. DEC/Test Manager submitted the specified collection for batch processing.

SUBSYSONLY, EXIT can only be used to leave the subsystem level

Explanation. The EXIT command is valid only when you are using DEC/Test Manager as a subsystem or when you are in the Review subsystem.

SUCCEEDED, the comparison for the test 'name' succeeded

Explanation. The result file for the specified test matched its benchmark file.

TESTNOTRUN, test 'name' was not run

Explanation. Execution of the collection stopped before all tests had executed.

User Action. See Chapter 5 for instructions for reviewing a partially run collection and for reexecuting the tests which did not execute.

TIMEORDER, BEFORE and SINCE time values cannot be resolved

Explanation. You specified an incorrect sequence of time values for the /BEFORE and /SINCE qualifiers. (The /SINCE qualifier value must indicate a time prior to that indicated by the /BEFORE qualifier value.)

User Action. Verify that you are entering the correct values, then reenter the command.

TOODEEP, eighth-level directory 'directory_spec' one level too deep

Explanation. You specified an eighth-level directory on the CREATE LIBRARY command which is the deepest directory level RMS allows. You cannot create a DEC/Test Manager library in an eighth-level directory because DEC/Test Manager cannot then create collection subdirectories.

User Action. Create another directory no deeper than a seventh-level directory.

TOOLONG, 'name' is too long, maximum of 'number' characters

Explanation. The specified expression is too long.

User Action. Reenter the command with a shorter expression.

TRMCTRL, control characters must be in range ^A through ^Z

Explanation. You specified a termination character as an out-of-range control character. Valid control characters include those between CTRL/A and CTRL/Z, inclusive.

User Action. Correctly specify the termination character.

TRMRANGE, termination character must be in range 0-255

Explanation. You specified a termination character that was either greater than 255 or negative. Valid values include decimal ASCII values between 0 and 255, inclusive.

User Action. Correctly specify the termination character.

TRMSYNTAX, specify termination character as single character or ^x or decimal ASCII value

Explanation. You specified an invalid termination character. Valid termination characters include single characters, control characters between CTRL/A and CTRL/Z, inclusive, and decimal ASCII values between 0 and 225, inclusive.

User Action. Correctly specify the termination character.

TRYAGNLAT, please try again later

Explanation. Your library is currently locked by another user.

User Action. Wait and reenter the command later.

TTY_READ, error reading from tty: device

Explanation. An error occurred while reading from your terminal.

User Action. See the secondary messages for more information.

TTY_WRITE, error writing to tty: device

Explanation. An error occurred while writing to your terminal.

User Action. See the secondary messages for more information.

UNDEFLIB, DEC/Test Manager library is now undefined

Explanation. You do not have a current DEC/Test Manager library.

User Action. Enter a SET LIBRARY command to establish a current DEC/Test Manager library.

UNKNOWN_SEQ, unknown control or escape sequence 'string'

Explanation. Your program output an ASCII control character or an escape sequence that DEC/Test Manager did not understand. DEC/Test Manager could not determine its effect on your terminal screen.

User Action. Examine your SESSION file to find the control character or escape sequence DEC/Test Manager did not understand. Verify that your program is outputting the correct escape sequences for the terminal on which it is running. For example, is it sending a VT100 escape sequence to a VT52 terminal? Verify that you are running your program and test on an appropriate terminal that is supported by DIGITAL. Verify that you are not using a terminal feature which this version of DEC/Test Manager does not support. Verify that you are not sending REGIS or SIXEL codes to your terminal.

UNMATQUOT, unmatched quote character in expression 'string'

Explanation. DEC/Test Manager detected a missing closing quote character (' or ").

User Action. Reenter the command with a correctly quoted string.

UNSUCCESS, the comparison for the test 'name' was unsuccessful

Explanation. The result file for the specified test did not match its benchmark file.

UNSUPFRMT, format of file 'name' not supported by COMPARE

Explanation. One of the files DEC/Test Manager is attempting to compare is not a text file.

UPDATED, the benchmark for test 'name' has been updated

Explanation. DEC/Test Manager replaced the current benchmark file with the result file.

UPDATERR, error in attempt to update the benchmark file for 'result-description-name'

Explanation. DEC/Test Manager could not replace the benchmark file with the result file, or DEC/Test Manager could not delete the result file.

User Action. See the secondary message for more information.

UPDATES, 'count' updates completed

Explanation. DEC/Test Manager updated the specified benchmark files.

UPDNOTNEC, update not necessary for 'result-description-name'

Explanation. You tried to update the benchmark file for a successful or updated test.

USERRECOVER, use DEC/Test Manager VERIFY/RECOVER

Explanation. Your library is in an inconsistent state.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

USESETLIB, use DEC/Test Manager SET LIBRARY

Explanation. Your DEC/Test Manager library is undefined.

User Action. Use the SET LIBRARY command to select a library and reenter the command.

VALREQUIRED, value required for variable 'name' in this context

Explanation. When you include the /VARIABLE qualifier on the CREATE COLLECTION command, you must include a variable-value for every variable-name you list.

User Action. Specify a variable-value for every variable-name you include.

VARCONFLICT, cannot change variable 'name' from 'null string/non-null numeric' to null numeric.

Explanation. You cannot create a numeric symbol variable with a null value.

User Action. Examine the use of this variable and determine whether the attempted operation makes sense for this variable.

VARNOVAL, 'name' variable has no default value. Must associate one with this test

Explanation. The specified variable does not have a default value.

User Action. Use the MODIFY VARIABLE command to associate a default value with the variable.

VERARC, archive list verified

Explanation. This phase of the VERIFY command completed successfully.

VERCOL, collection list verified

Explanation. This phase of the VERIFY command completed successfully.

VERCOLDIR, collection directory structure verified

Explanation. This phase of the VERIFY command completed successfully.

VERFRE, free space list verified

Explanation. This phase of the VERIFY command completed successfully.

VERGRO, group list verified

Explanation. This phase of the VERIFY command completed successfully.

VERHEAD, user header information verified

Explanation. This phase of the VERIFY command completed successfully.

VERIFIED, DEC/Test Manager library 'directory-spec' verified

Explanation. Your DEC/Test Manager library has been successfully verified.

VERNOTALL, explicit version numbers not allowed in input file specifications

Explanation. You cannot include a version number on a file specification entered as a parameter to a DEC/Test Manager command.

User Action. Remove the version number references from all file specifications and reenter the command.

VERSPACE, contiguous space verified

Explanation. This phase of the VERIFY command completed successfully.

VERSTR, string list verified

Explanation. This phase of the VERIFY command completed successfully.

VERTD, test description list verified

Explanation. This phase of the VERIFY command completed successfully.

VERVAR, variables list verified

Explanation. This phase of the VERIFY command completed successfully.

WAITFRERR, unable to wait for a system event flag to set

Explanation. A system service failed while waiting for an event flag.

User Action. Check your quotas and SYSGEN parameters. Consult with your system manager.

WAITING, DEC/Test Manager library 'name' is still in use

Explanation. Someone else is still using your library and your command cannot yet be executed.

User Action. DEC/Test Manager will automatically resume execution of your command when the library is free.

WLDNOTALLOWED, wildcard in expression 'name' not allowed

Explanation. You cannot include a wildcard character (* or %) in the specified parameter.

User Action. Reenter the command with a valid parameter.

YOUDEL, you must manually delete the collection files

Explanation. DEC/Test Manager could not delete all the files in the specified collection.

User Action. Use the DCL DELETE command to delete the remaining files.

YOUDELCRE, you must manually delete and recreate this collection

Explanation. The original CREATE COLLECTION command was more than 255 characters long. Due to CLI restrictions, you cannot currently use the RECREATE command for this collection.

User Action. Use the DELETE COLLECTION command to manually delete the existing collection. Then reenter the original CREATE COLLECTION command line.

ZLENBLO, a zero length block was found during pass 2

Explanation. The library structure contains an error.

User Action. Use the VERIFY/RECOVER command to restore your library to a usable state.

SESSION Files and SESSION File Editing

This appendix describes SESSION files and INPUT files and discusses how they are used. It presents information on the following:

- The format of SESSION files
- The format of INPUT files
- Editing SESSION files by means of INPUT files that you can read and edit
 - Creating an INPUT file from an existing SESSION file
 - Creating an INPUT file with an editor
 - Creating a SESSION file from one or more INPUT files

A SESSION file contains a record of an interactive terminal session recorded for the purpose of interactively testing a program with DEC/Test Manager. A SESSION file has the default file type SESSION and contains a description of the type of terminal on which you recorded the terminal session, a record of all characters you input during the terminal session, a record of all characters the computer output during the terminal session, and additional control and timing information.

An INPUT file contains a text representation of all or part of an interactive terminal session. You can read INPUT files and edit them with the editor of your choice. An INPUT file has the default file type INP and contains a record of all characters you input during an interactive terminal session and **special strings**, which are text representations of nonprinting text and recording functions contained in SESSION files. An INPUT file does not contain a record of characters output by the computer during the terminal session.

CAUTION

DEC/Test Manager is guaranteed to work only with unmodified SESSION files it creates and with SESSION files produced from properly generated INPUT files. If you write programs to create SESSION files, or if you edit or otherwise modify SESSION files, do so using INPUT files as described in this appendix.

SESSION files created or modified outside of DEC/Test Manager are not guaranteed to be upwardly compatible with future versions of DEC/Test Manager. SESSION files created from properly generated INPUT files are guaranteed to be upwardly compatible.

The SESSION file format described in this appendix may be superseded.

B.1 SESSION Files

A SESSION file consists of one record containing a 12-byte terminal characteristics block followed by a sequence of records, each beginning with a one-character record type. The record structure of a SESSION file is described in Section B.1.2.1.

B.1.1 Sample SESSION File

The following example shows the SESSION file resulting from recording a terminal session where you enter the DCL SHOW DEFAULT command. Entering the DTM CREATE TEST_DESCRIPTION/RECORD command to record the SESSION file results in the following terminal session:

```

$ DTM CREATE TEST_DESCRIPTION/RECORD
_test name: test1
_Remark: Creating a simple SESSION file
%DTM-I-DEFAULTED, benchmark file name defaulted to TEST1.BMK
%DTM-I-DEFAULTED, template file name defaulted to TEST1.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.
```

```
⌘ SHOW DEFAULT
DRA1: [PROJECT.MYLIB]
⌘ ^P^P
```

^P

```
%DTM-I-BMK_SAVED, benchmark has been saved in file
DRA1: [project.dtmlib]test1.bmk;1
%DTM-S-RECORDED, test TEST1 has been successfully recorded in
DRA1: [project.mylib]test1.session
%DTM-S-CREATED, test description test1 created
⌘
```

The following SESSION file, TEST1.SESSION, was produced by DEC/Test Manager for this interactive terminal session:

```
① B'P^@<XAO>~S^A^X^D^P<X82>)
② ! DTM V2.1 RECORD V2.1
O<CR>
O<LF>
O^@
O$
O
③ O^A
Is
Os
Ih
Oh
Io
Oo
Ow
TO :00:01.0
I
O
Id
Od
Ie
Oe
If
Of
Ia
Oa
Iu
Ou
Il
Ol
It
Ot
I<CR>
O<CR>
O<LF>
O<CR>
O DRA1: [PROJECT.MYLIB]
```

```
0<CR>
0<LF>
0<CR>
0<CR>
0<LF>
0^@
0$
0
```

③ 0^A

The indicated records contain:

- ① The terminal characteristics block
- ② A comment record
- ③ A CTRL/A indicating that DEC/Test Manager will compare this screen

B.1.2 SESSION File Structure

Because both timing and system load can affect the performance of the system and the application being tested, two or more SESSION files recorded independently may result in completely different, yet valid, SESSION files. This occurs even though you typed exactly the same keystrokes when recording both terminal sessions. Thus, performing a source comparison on two SESSION files yields no useful information.

Because most VAX/VMS terminal drivers are full duplex, when dealing with SESSION files you should consider input and output to be asynchronous events. You can be typing input while the computer is simultaneously writing output.

Input and output may appear mixed in a SESSION file. For example, if you were to type input such as ABCDEF, first you would type A and a few milliseconds later the computer would output A. Then you would type B and the computer would output B, and so on. The corresponding SESSION file contains a record showing that you typed A, followed by records showing that the computer output A, you typed B, the computer output B, and so on. This sequence is shown in the following partial SESSION file.

.
.
.
Ia
Oa
Ib
Ob
Ic
Oc
Id
Od
Ie
Oe
If
Of
.
.

Both the speed at which you type and system response time affect the SESSION file. If you type quickly or if system response is slow, for example, the SESSION file will be different from a SESSION file recorded under different conditions. For the same input (ABCDEF), another SESSION file recorded under different conditions might contain a record showing that you entered AB, followed by records showing the computer outputting A, you inputting C, then the computer outputting BC, and so on. This sequence is shown in the following partial SESSION file:

.
.
.
Iab
Oa
Ic
Obc
Ide
Ode
If
Of
.
.

Although you typed the same input, ABCDEF, both times and the computer echoed back the same letters, the two SESSION files are completely different. SESSION files recorded under different circumstances will vary, but all SESSION files are equally valid.

The SESSION file will be further varied if the program outputs something to the screen other than what you type. For example, the program might output X whenever you type A or it might output nothing at all. Asynchronous events, for example, entering CTRL/T or a broadcast write that occurs while the computer is echoing your input, will change the SESSION file.

The terminal driver may also arbitrarily buffer up text or split it into groups before outputting it. Even though your program may have issued a 6-character QIO to output ABCDEF, the terminal driver may output the text as two groups, ABCD followed by EF; or as 6 separate characters; or as a single 6-character string. The terminal driver may even output the text appended to some previously entered text which has not yet been output. Each elementary operation of the terminal driver results in a separate record being written to the SESSION file. Thus, each record in the SESSION file describes a single terminal driver operation.

DEC/Test Manager also writes additional timing and control information to the SESSION file.

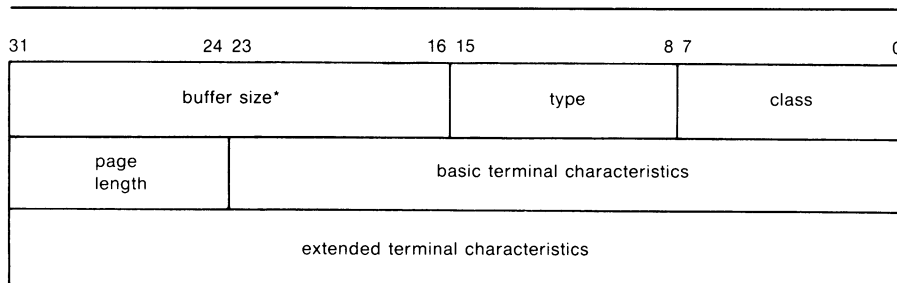
B.1.2.1 Record Structure of SESSION Files

The record structure of a SESSION file is extremely important. You must not change it except as described in this section.

The First Record—The Terminal Characteristics Block

The first record in a SESSION file is a 12-byte (12-character) **terminal characteristics block**. This block of information describes the type of terminal on which the terminal session was recorded and the characteristics of that terminal. The terminal characteristics block is described in the *VAX/VMS I/O User's Reference Manual: Part 1* and is shown in Figure B-1.

Figure B-1: Format of the Terminal Characteristics Block



*page width

P2 = 12

ZK-693-82

The 12 bytes in the record are stored low order to high order as three longwords. The record conforms to the structure returned by a sensemode terminal QIO with a P2 parameter of 12. The first byte has the value DC\$_TERM.

CAUTION

You must not change the terminal characteristics block in any way. Any change to this record may invalidate the entire file. Do not add bytes to this record or delete bytes from it. This record must contain exactly 12 bytes.

Succeeding Records

All succeeding records in the SESSION file begin with a one-byte **record type** which describes the contents of the record. The record type is a number, usually represented by its corresponding ASCII character. For example, the decimal number 65 is referred to by the letter A. The formats for all possible record types are shown in Table B-1.

Table B-1: SESSION File Record Types

Record Type	Description
B	(BEGIN_COMPARE) Restarts automatic screen compare terminated by a previous E record. By default, when an input point is reached DEC/Test Manager automatically marks the current screen for comparison with the corresponding screen in the benchmark file.
C	(COMPARE_SCREEN) Forces DEC/Test Manager to mark the current screen for comparison while automatic screen comparison is turned off.
D	(DELAY) Contains a standard VAX/VMS delta time specification. This record type is reserved and its use is not recommended. Currently, DEC/Test Manager interprets this as a W record type.
E	(END_COMPARE) Terminates automatic screen compare. By default, when an input point is reached DEC/Test Manager automatically marks the current screen for comparison with the corresponding screen in the benchmark file.
I	(INPUT) Contains characters you input, that is, characters you type at the terminal. This record usually contains only one character. These characters do not automatically echo to the screen unless the terminal is set in half-duplex mode. This input is usually echoed in a subsequent O record.
O	(OUTPUT) Labels the record as containing characters output by the computer and displayed on the terminal.
T	(TIMING) Contains a standard VAX/VMS delta-time specification. This time interval represents the clock time elapsed between the previous I (INPUT) record and the next I record.
W	(WAIT) Contains a standard VAX/VMS delta time specification. It produces a pause of the specified length in the input stream when the terminal session is played back.
!	(BEGIN_COMMENT) Contains a comment. This record is ignored.
0	(Null or OUTPUT) This record type is supported for compatibility and its use is not recommended. DEC/Test Manager interprets this as an O record.
1	(CTRL/A or INPUT) This record type is supported for compatibility and its use is not recommended. DEC/Test Manager interprets this as an I record.

The T and W (TIMING and WAIT) records contain standard VAX/VMS delta time specifications. Currently, the D (DELAY) record type is interpreted as a W record type. A time interval of 2.3 seconds would appear in a T record as follows:

```
TO :00:02.3
```

It would appear in a W record as follows:

```
WO :00:02.3
```

In T records, the time interval represents the elapsed clock time between the previous and next I records. DEC/Test Manager does not currently interpret T records. DEC/Test Manager does not normally record timing intervals of less than one second.

When DEC/Test Manager replays a terminal session, T records cause a time delay on the input stream when DEC/Test Manager replays the session at the speed at which it was recorded. In all other cases, T records are ignored.

In W and D records, this time interval causes a pause in the input stream of the specified duration. In other words, when DEC/Test Manager replays the terminal session, it ignores T records but interprets W records.

B.1.2.2 Modifying Session Files Directly

Seemingly harmless changes, such as changing the case of letters in a SESSION file to all uppercase or all lowercase may invalidate the entire file. This action changes the case and, therefore, the meaning of characters inside control and escape sequences.

Passing a SESSION file through some editors (such as TECO) may change the record structure, or add or remove new line characters such as LINE FEEDS or RETURNS. These changes invalidate the SESSION file.

In general, splitting a record or combining two consecutive records may invalidate the entire SESSION file. If you modify the records in a SESSION file directly, observe the guidelines in the following sections. It is recommended that you modify SESSION files indirectly using INPUT files. Input files are described in Section B.2.

Modifying O Records

Two consecutive O records can be combined into a single O record. When you do this, drop the O character from the second record.

You can split an O record into multiple O records. When you do this, begin each new record with an O record type.

You must preserve the number of CTRL/A characters in the output stream. A single CTRL/A character represents a point where the program is requesting input and where DEC/Test Manager compares screens. Two consecutive CTRL/A characters represent an input point where automatic screen comparison is suppressed.

The terminal driver inserts a CTRL/A character into the output stream whenever the program issues an input QIO. If the program prompts for the input, the CTRL/A character appears after the prompt. Additionally, DEC/Test Manager sometimes inserts CTRL/A characters into the SESSION file to mark input points for comparison.

Modifying I Records

Two consecutive I records can be combined into a single I record. When you do this, drop the I character from the second record.

You may split an I record into multiple I records. When you do this, begin each new record with an I record type.

Modifying B, C, and E Records

The B, C, and E (BEGIN_COMPARE, COMPARE_SCREEN, and END_COMPARE) records can occur in the SESSION file only as the next record immediately following an O record that ends with a CTRL/A character. There is one exception to this: a B, C, or E record can be the first noncomment record in the file immediately following the terminal characteristics block. If a B, C, or E record occurs in a SESSION file, that file must also contain at least one O record. The number of CTRL/A characters in the file must agree with the number of CTRL/A characters output by the interactive terminal session when it is run with the PLAY command.

Modifying D, T, and W Records

D, T, and W records can be modified or deleted to change or remove timing information. When modifying this record, you must enter the delta time in a valid format as specified in the *VAX/VMS System Services Reference Manual*. This is not necessarily in the format that DEC/Test Manager uses. Do not omit any required punctuation.

Note that the behavior of some programs varies depending on the speed of user input. Removing or modifying timing information for such programs could adversely affect the way these programs run.

B.2 INPUT Files

INPUT files contain a text representation of an interactive terminal session as recorded by DEC/Test Manager in a SESSION file. INPUT files contain the following information:

- Your input for the terminal session—the characters you typed when you recorded the terminal session
- All nonprinting characters and recording functions represented as special strings

INPUT files can be created in several ways. You can record an INPUT file from an existing SESSION file using the EXTRACT command, or you can generate an INPUT file from scratch using the text editor of your choice. You can also use a combination of these techniques. For example, you can edit an existing test script to reformat it as an INPUT file.

B.2.1 Sample INPUT File

The following example shows the INPUT file generated from the SESSION file TEST1.SESSION recorded in Section B.1.1. Enter the EXTRACT command to generate the INPUT file. Supply the name of the SESSION file from which the INPUT file is to be extracted and the name for the INPUT file.

```
$ DTM EXTRACT
_session file: TEST1.SESSION
_input file: TEST1.INP
```

```
%DTM-S-EXTRACTED, input file DRA1:[project.dtmlib]test1.inp created
$
```

The following INPUT file, TEST1.INP, was produced by the EXTRACT command:

```
show default{<CR>}
```

The INPUT file TEST1.INP contains the text entered during the terminal session (the DCL SHOW DEFAULT command) and the special string { <CR> } that terminates the entered text. Notice that output supplied by the computer is not included in the input file.

This INPUT file can be edited and then used to generate a new SESSION file.

B.2.2 Special Strings

All nonprinting characters and recording functions found in SESSION files are replaced in INPUT files by **special strings** which have the following format:

```
{special-string}
```

These special strings are text representations for the nonprinting characters and recording functions. They are enclosed within braces ({ and }).

B.2.2.1 Types of Special Strings Recognized by DEC/Test Manager

DEC/Test Manager recognizes the following types of special strings:

- Control and nonprinting characters—mnemonic control character names and decimal integer values for control characters available in both 7-bit and 8-bit environments
- Common names for nonprinting characters
- 8-bit control characters—mnemonic control character names and decimal integer values for control characters available only in 8-bit environments
- Key names
 - names written on keyboard keys (with underscores substituted for spaces)
 - names for the arrow keys
 - names for the editing keys
 - names for the keypad keys
 - names for the function keys
- Names for the recording functions

When you extract an INPUT file from a SESSION file, DEC/Test Manager translates each nonprinting character and recording function in the SESSION file into the appropriate special string. When you generate a SESSION file from an INPUT file, DEC/Test Manager retranslates the special strings. The translations performed when extracting an INPUT file from a SESSION file are listed in Table B-2. The translations performed when recording a SESSION file from an INPUT file are listed in Tables B-3 through B-9.

Control Characters and Common Names for Nonprinting Characters

DEC/Test Manager recognizes the following formats for control characters and nonprinting characters:

- Special strings for control characters of the forms {CTRL/x} and {^x}.
For example, DEC/Test Manager interprets both {CTRL/A} and {^A} as the same control character.
- Special strings for all mnemonic control character names listed in the ASCII 7-bit and 8-bit character tables surrounded by angle brackets < and > .

For example, DEC/Test Manager interprets { <SOH> } and { <IND> }.

- Special strings for all integer decimal values for control characters listed in the ASCII 7-bit and 8-bit character tables.

For example, DEC/Test Manager interprets {27} as ESC.

DEC/Test Manager ignores leading zeros on integer decimal values.

- Special strings for common names for nonprinting characters:

```
{ <BACK_SPACE> }  
{ <DELETE> }  
{ <ENTER> }  
{ <ESC> }  
{ <ESCAPE> }  
{ <FORM_FEED> }  
{ <LINE_FEED> }  
{ <PAGE> }  
{ <RETURN> }  
{SPACE}, { <SP> }, {32}, and regular spaces { } are all recognized  
as the space character  
{ <TAB> }
```

Tables B-3 and B-4 list the special string translations that DEC/Test Manager performs for control characters. Table B-3 also lists the special string translations for common names for nonprinting characters. Also check your terminal manual for ASCII 7-bit and 8-bit tables.

Key Names

DEC/Test Manager recognizes special strings for the names written on the keyboard keys with underscores (_) substituted for spaces. For example, DEC/Test Manager recognizes {RETURN}, {LINE_FEED}, and {PF1}.

DEC/Test Manager recognizes special strings for the arrow keys by name. For example, DEC/Test Manager recognizes both {UP_ARROW} and {UP}.

DEC/Test Manager recognizes special strings for the editing keys name. For example, DEC/Test Manager recognizes {REMOVE}, {NEXT_SCREEN}, and {NEXT}.

DEC/Test Manager recognizes special strings for the keypad keys by name. For example, DEC/Test Manager recognizes {KP3}, {MINUS}, and {ENTER}.

DEC/Test Manager recognizes special strings for the function keys by name. For example, DEC/Test Manager recognizes both {F12} and {BS}.

Tables B-5 through B-8 list the special string translations that DEC/Test Manager performs for key names. Table B-5 lists the special string translations for the function key codes. Table B-6 lists the special string translations for the editing key codes. Table B-7 lists the special string translations for the keypad key codes. Table B-8 lists the special string translations for the arrow key codes.

Recording Functions

DEC/Test Manager recognizes the following special strings for the recording functions:

- {BEGIN_COMMENT} and {END_COMMENT}
- {BEGIN_COMPARE} and {END_COMPARE}
- {COMPARE_SCREEN}
- {DELAY}
- {WAIT}

You should use these special strings in your INPUT files to avoid making the INPUT file dependent on a particular termination character. For example, if you enter {CTRL/P}C in an INPUT file to mark a screen for comparison, the INPUT file and all SESSION files generated from it must be used in conjunction with the termination character CTRL/P. But if you enter {COMPARE_SCREEN} to mark a screen for comparison, the INPUT file and all SESSION files generated from it can be used in conjunction with any termination character.

Table B-9 lists the special string translations that DEC/Test Manager performs for the recording functions.

B.2.2.2 Using Special Strings in INPUT Files

DEC/Test Manager's interpretation of special strings in INPUT files is not case sensitive. You can use uppercase or lowercase characters, or a combination of the two when you enter a special string in an INPUT file.

When processing INPUT files, DEC/Test Manager does not interpret the end of a record as the end of input. Therefore, you must be careful to enter a special string corresponding to a carriage return, (for example, { <CR> }) at the end of any input normally terminated with a carriage return.

To include text enclosed within braces ({ and }) in the INPUT file, enter double braces around the text, (for example, {{ text }}). When processing the INPUT file, DEC/Test Manager will translate the double braces to single braces; DEC/Test Manager will not interpret the text contained within the braces as a special string.

Use the special string equivalents for the recording functions when writing an INPUT file. This avoids building a dependency on a particular termination character into your INPUT file.

When you enter a comment in an INPUT file, enclose the comment between the {BEGIN_COMMENT} and {END_COMMENT} special strings. Enter the {BEGIN_COMMENT} and {END_COMMENT} special strings on lines of their own because DEC/Test Manager ignores the remainder of the line following these special strings. You must begin each line of comment text with the comment character (!).

Do not nest INPUT files. DEC/Test Manager will ignore INSERT recording functions (CTRL/P I) when they occur in INPUT files.

B.3 Editing SESSION Files Using INPUT Files

The process of editing an interactive terminal session is simplified if you do so using an intermediate text file called an INPUT file. You can extract an INPUT file from an existing SESSION file, you can create an input file from scratch with a text editor, or you create an INPUT file by editing an existing test script. When the INPUT file is complete and correctly formatted, you can use it to record a SESSION file.

B.3.1 Extracting an INPUT File from a SESSION File—EXTRACT

The EXTRACT command extracts an INPUT file from an existing SESSION file without altering the SESSION file. The format is:

```
DTM EXTRACT session-file-spec [input-file-spec]
```

The session-file-spec is the file specification of the SESSION file from which DEC/Test Manager is to extract an INPUT file. If you specify a file-name for the SESSION file without specifying a file-type, the file-type defaults to SESSION. The input-file-spec is the file specification for the INPUT file being created. If you do not specify an input-file-spec, the file specification defaults to session-file-name.INP. If you specify an input-file-name without specifying a file-type, the file-type defaults to INP. Both files can be stored in DEC/CMS libraries. The EXTRACT command will fetch the SESSION file from a DEC/CMS library, but it will not place the INPUT file in a CMS library. You must place the INPUT file in a DEC/CMS library manually yourself by issuing DEC/CMS commands. If you subsequently record a SESSION file from an INPUT file stored in a DEC/CMS library, you must first fetch the INPUT from the DEC/CMS library by issuing the appropriate DEC/CMS commands. You must do this manually yourself.

The EXTRACT command takes the /[NO]LOG and /TERMINATION_CHARACTER qualifiers. The /TERMINATION_CHARACTER qualifier specifies the termination character that DEC/Test Manager is to use when translating the recording functions in the SESSION file to special strings in the INPUT file. If the interactive terminal session you are recording does not use the default termination character (CTRL/P), you need not specify a different termination character. If the interactive terminal session you are recording uses CTRL/P for its own purposes, you must specify a different termination character.

When DEC/Test Manager extracts the INPUT file from the SESSION file the following occurs:

- All input in the SESSION file is written to the INPUT file.
- All nonprinting characters and recording functions are translated to special strings—text delimited by braces ((and)).
- Any braces appearing in the text of the SESSION file are doubled in the INPUT file.

The following example extracts the INPUT file TEST1_A.INP from the SESSION file TEST1.SESSION:

```
‡ DTM EXTRACT TEST1.SESSION TEST1_A.INP

%DTM-S-EXTRACTED, input file DRA1:[project.dtmlib]test1_a.inp created
‡
```

B.3.2 Creating an INPUT File with a Text Editor

You can create an INPUT file using the text editor of your choice. See Section B.2.2.2 for information on using special strings in INPUT files.

The following example shows a sample INPUT file created with a text editor:

```
{BEGIN_COMMENT}
! This is a sample INPUT file. When used, it calls up EDT
! to create a file, enters some text into the buffer, and
! walks around that text, finally quitting without saving
! the file.
{END_COMMENT}
edit/edt sample.tmp{<CR>}
change{<CR>}
{BEGIN_COMMENT}
! Enter the text into the buffer.
{END_COMMENT}
This is the first line of the file.{<CR>}
This is the
second line of the file.{<CR>}
{UP_ARROW}{UP}{KP2}
{SPACE}This is more of the first line.
{CTRL/Z}
quit{<CR>}
```

B.3.3 Recording a SESSION File from an INPUT File

You can record a SESSION file from an INPUT file in two ways:

- By using the /INPUT qualifier on a CREATE or MODIFY TEST_DESCRIPTION command
- By entering the INSERT recording function (CTRL/P I) while recording an interactive terminal session

B.3.3.1 Using the CREATE or MODIFY TEST_DESCRIPTION/INPUT Command

The /INPUT qualifier on the CREATE or MODIFY TEST_DESCRIPTION command specifies that DEC/Test Manager record a new SESSION file by initiating an interactive terminal session and taking input from the specified INPUT file. If you do not also specify the /RECORD qualifier, DEC/Test Manager terminates the interactive terminal session when the INPUT file is exhausted. If you include both the /INPUT and /RECORD qualifiers, DEC/Test Manager leaves the terminal in record mode when the INPUT file is exhausted. You can then continue the terminal session interactively and terminate it by entering the termination character (CTRL/P) twice.

The following example initiates an interactive terminal session to create SESSION file TEST2.SESSION, takes all input from INPUT file TEST2.INP, and terminates the terminal session when TEST2.INP is exhausted.

```
$ DTM CREATE TEST_DESCRIPTION/INPUT=TEST2.INP
_test name: TEST2
_Remark: Recording TEST2.SESSION from TEST2.INP
%DTM-I-DEFAULTED, benchmark file name defaulted to TEST2.BMK
%DTM-I-DEFAULTED, template file name defaulted to TEST2.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...

$ show default
  DRA1: [PROJECT.MYLIB]

~P

%DTM-I-BMK_SAVED, benchmark has been saved in file
  DRA1: [project.dtmlib]test2.bmk;1
%DTM-S-RECORDED, test TEST2 has been successfully recorded in
  DRA1: [project.mylib]test2.session
%DTM-S-CREATED, test description test2 created
$
```

The following example initiates an interactive terminal session to create SESSION file SAMPLE_TEST.SESSION, takes input from INPUT file SAMPLE_TEST.INP until the file is exhausted, and leaves the terminal in record mode.

```

$ DTM CREATE TEST_DESCRIPTION/RECORD/INPUT=SAMPLE_TEST.INP
_test name: sample_test
_Remark: Recording a sample SESSION file
%DTM-I-DEFAULTED, benchmark file name defaulted to SAMPLE_TEST.BMK
%DTM-I-DEFAULTED, template file name defaulted to SAMPLE_TEST.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.

$ show default
  DRA1: [PROJECT.MYLIB]
$ show time
$ ^P^P
^P

%DTM-I-BMK_SAVED, benchmark has been saved in file
  DRA1: [project.dtmlib]sample_test.bmk;1
%DTM-S-RECORDED, test SAMPLE_TEST has been successfully recorded in
  DRA1: [project.mylib]sample_test.session
%DTM-S-CREATED, test description sample_test created
$
```

B.3.3.2 Using the INSERT Recording Function—CTRL/P I

When the INSERT recording function CTRL/P I is entered during an active recording session, it specifies that DEC/Test Manager take input from the specified INPUT file. When you enter the INSERT recording function, DEC/Test Manager prompts you for the file specification for a single INPUT file. DEC/Test Manager then takes input from the specified INPUT file and returns control to the terminal when the INPUT file is exhausted.

```

$ DTM CREATE TEST_DESCRIPTION/INPUT=SAMPTEST.INP
_test name: SAMPTEST
_Remark: Recording SAMPTEST.SESSION from SAMPTEST.INP
%DTM-I-DEFAULTED, benchmark file name defaulted to SAMPTEST.BMK
%DTM-I-DEFAULTED, template file name defaulted to SAMPTEST.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.
```

```

.
.
.
^PI
```

```
_Input file: SAMPTEST2.INP
.
.
.
^P^P
^P
%DTM-I-BMK_SAVED, benchmark has been saved in file
  DRA1:[project.dtmlib]samptest.bmk;1
%DTM-S-RECORDED, test SAMPTEST has been successfully recorded in
  DRA1:[project.mylib]samptest.session
%DTM-S-CREATED, test description samptest created
$
```

During a single terminal session, you can read input from multiple INPUT files sequentially. The INPUT files cannot be nested. DEC/Test Manager ignores any INSERT recording functions in an INPUT file.

B.3.3.3 Terminal Characteristics

The process of creating an INPUT file is not terminal specific. When you extract an INPUT file from a SESSION file, all control codes and other nonprinting characters are translated to special strings regardless of whether they have meaning to the terminal you are using to perform the translation.

The process of creating a SESSION file from an INPUT file is terminal specific. When you record a SESSION file, DEC/Test Manager translates all special strings back to control codes and other nonprinting characters based on the terminal characteristics for the recording terminal. If DEC/Test Manager encounters a special string that it cannot translate for the recording terminal, the braces are stripped off the special string and the characters representing the untranslated special string are printed in the SESSION file. They are also displayed on the terminal screen along with an error message.

When you record a SESSION file, the terminal characteristics for the recording terminal become the first record of that SESSION file. That SESSION file is guaranteed to run on terminals of the same type as the recording terminal. The SESSION file may or may not run on other terminal types.

NOTE

You may encounter problems rerecording a SESSION file on a VT100 series terminal if the SESSION file was originally recorded on a VT200 series terminal. Problems will occur if the original SESSION file contains control codes which are restricted to use in an 8-bit compatible environment. These codes are listed in Table B-4. DEC/Test Manager cannot translate these 8-bit control codes for use in a 7-bit environment.

If you record an interactive terminal session on a VT100 series terminal and extract an INPUT file from this SESSION file, you will be able to successfully rerecord the edited terminal session on either a VT100 or VT200 series terminal.

B.3.3.4 Type-Ahead

Anything you type on your terminal while input is being taken from an INPUT file will have no immediate affect on the terminal session. All or part of what you type may be stored as type-ahead and may appear when the INPUT file is exhausted and control is returned to the terminal.

B.4 Translation Tables

Table B-2 describes translation of nonprinting characters and control codes when an INPUT file is extracted from a SESSION file.

**Table B-2: Translation of Nonprinting Characters and Control Codes
When Extracting an INPUT File from a SESSION File**

Code in SESSION File		Translated Special String in INPUT File		
Mnemonic	Recording Function	8-bit Control String	Escape Sequence	Special String
—	—	<CSI> A and <SS3> A	<ESC> A and <ESC> [A and <ESC> O A	{UP_ARROW}
—	—	<CSI> B and <SS3> B	<ESC> B and <ESC> [B and <ESC> O B	{DOWN_ARROW}
—	—	<CSI> C and <SS3> C	<ESC> C and <ESC> [C and <ESC> O C	{LEFT_ARROW}
—	—	<CSI> D and <SS3> D	<ESC> D and <ESC> [D and <ESC> O D	{RIGHT_ARROW}
—	—	<SS3> P	<ESC> P and <ESC> O P	{PF1}
—	—	<SS3> Q	<ESC> Q and <ESC> O Q	{PF2}
—	—	<SS3> R	<ESC> R and <ESC> O R	{PF3}
—	—	<SS3> S	<ESC> S and <ESC> O S	{PF4}
—	—	<SS3> l	<ESC> ? l and <ESC> O l	{COMMA}
—	—	<SS3> m	<ESC> ? m and <ESC> O m	{MINUS}
—	—	<SS3> n	<ESC> ? n and <ESC> O n	{PERIOD}
—	—	<SS3> M	<ESC> ? M and <ESC> O M	{ENTER}
—	—	<SS3> p	<ESC> ? p and <ESC> O p	{KP0}

Table B-2 (Cont.): Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File

Code in SESSION File				Translated Special String in INPUT File
Mnemonic	Recording Function	8-bit Control String	Escape Sequence	Special String
—	—	<SS3> q	<ESC> ? q and <ESC> O q	{KP1}
—	—	<SS3> r	<ESC> ? r and <ESC> O r	{KP2}
—	—	<SS3> s	<ESC> ? s and <ESC> O s	{KP3}
—	—	<SS3> t	<ESC> ? t and <ESC> O t	{KP4}
—	—	<SS3> u	<ESC> ? u and <ESC> O u	{KP5}
—	—	<SS3> v	<ESC> ? v and <ESC> O v	{KP6}
—	—	<SS3> w	<ESC> ? w and <ESC> O w	{KP7}
—	—	<SS3> x	<ESC> ? x and <ESC> O x	{KP8}
—	—	<SS3> y	<ESC> ? y and <ESC> O y	{KP9}
—	—	<CSI> 17~	<ESC> [17~	{F6}
—	—	<CSI> 18~	<ESC> [18~	{F7}
—	—	<CSI> 19~	<ESC> [19~	{F8}
—	—	<CSI> 21~	<ESC> [21~	{F10}
—	—	<CSI> 23~	<ESC> [23~	{F11}
—	—	<CSI> 24~	<ESC> [24~	{F12}
—	—	<CSI> 25~	<ESC> [25~	{F13}
—	—	<CSI> 26~	<ESC> [26~	{F14}
—	—	<CSI> 28~	<ESC> [28~	{F15}

Table B-2 (Cont.): Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File

Code in SESSION File			Translated Special String in INPUT File	
Mnemonic	Recording Function	8-bit Control String	Escape Sequence	Special String
—	—	<CSI> 29~	<ESC> [29~	{F16}
—	—	<CSI> 31~	<ESC> [31~	{F17}
—	—	<CSI> 32~	<ESC> [32~	{F18}
—	—	<CSI> 33~	<ESC> [33~	{F19}
—	—	<CSI> 34~	<ESC> [34~	{F20}
—	—	<CSI> 1~	<ESC> [1~	{FIND}
—	—	<CSI> 2~	<ESC> [2~	{INSERT_HERE}
—	—	<CSI> 3~	<ESC> [3~	{REMOVE}
—	—	<CSI> 4~	<ESC> [4~	{SELECT}
—	—	<CSI> 5~	<ESC> [5~	{PREV_SCREEN}
—	—	<CSI> 6~	<ESC> [6~	{NEXT_SCREEN}
<NUL>	—	—	—	{CTRL/@}
<SOH>	—	—	—	{CTRL/A}
<STX>	—	—	—	{CTRL/B}
<ETX>	—	—	—	{CTRL/C}
<EOT>	—	—	—	{CTRL/D}
<ENQ>	—	—	—	{CTRL/E}
<ACK>	—	—	—	{CTRL/F}
<BEL>	—	—	—	{CTRL/G}
<BS>	—	—	—	{ <BS> }
<HT>	—	—	—	{ <TAB> }
<LF>	—	—	—	{ <LF> }
<VT>	—	—	—	{CTRL/K}
<FF>	—	—	—	{ <FF> }
<CR>	—	—	—	{ <CR> }

Table B-2 (Cont.): Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File

Code in SESSION File				Translated Special String in INPUT File
Mnemonic	Recording Function	8-bit Control String	Escape Sequence	Special String
<SO>	—	—	—	{CTRL/N}
<SI>	—	—	—	{CTRL/O}
<DLE>	—	—	—	{CTRL/P}
<DC1>	—	—	—	{CTRL/Q}
<DC2>	—	—	—	{CTRL/R}
<DC3>	—	—	—	{CTRL/S}
<DC4>	—	—	—	{CTRL/T}
<NAK>	—	—	—	{CTRL/U}
<SYN>	—	—	—	{CTRL/V}
<ETB>	—	—	—	{CTRL/W}
<CAN>	—	—	—	{CTRL/X}
	—	—	—	{CTRL/Y}
<SUB>	—	—	—	{CTRL/Z}
<ESC>	—	—	—	{ <ESC> }
<FS>	—	—	—	{CTRL/\}
<GS>	—	—	—	{CTRL/}
<RS>	—	—	—	{CTRL/^}
<US>	—	—	—	{CTRL/_}
	—	—	—	{ }
<IND>	—	—	—	{ <IND> }
<NEL>	—	—	—	{ <NEL> }
<SSA>	—	—	—	{ <SSA> }
<ESA>	—	—	—	{ <ESA> }
<HTS>	—	—	—	{ <HTS> }
<HTJ>	—	—	—	{ <HTJ> }

Table B-2 (Cont.): Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File

Mnemonic	Code in SESSION File			Translated Special String in INPUT File
	Recording Function	8-bit Control String	Escape Sequence	Special String
<VTS>	—	—	—	{ <VTS> }
<PLD>	—	—	—	{ <PLD> }
<PLU>	—	—	—	{ <PLU> }
<RI>	—	—	—	{ <RI> }
<SS2>	—	—	—	{ <SS2> }
<SS3>	—	—	—	{ <SS3> }
<DCS>	—	—	—	{ <DCS> }
<PU1>	—	—	—	{ <PU1> }
<PU2>	—	—	—	{ <PU2> }
<STS>	—	—	—	{ <STS> }
<CCH>	—	—	—	{ <CCH> }
<MW>	—	—	—	{ <MW> }
<SPA>	—	—	—	{ <SPA> }
<EPA>	—	—	—	{ <EPA> }
<CSI>	—	—	—	{ <CSI> }
<ST>	—	—	—	{ <ST> }
<OSC>	—	—	—	{ <OSC> }
<PM>	—	—	—	{ <PM> }
<APC>	—	—	—	{ <APC> }
—	!+	—	—	{BEGIN_COMMENT}
—	!-	—	—	{END_COMMENT}
—	CTRL/P ¹ B	—	—	{BEGIN_COMPARE}
—	CTRL/P ¹ C	—	—	{COMPARE_SCREEN}

¹CTRL/P is used here only as an example. The actual value will be the termination character specified when the SESSION file was recorded.

Table B-2 (Cont.): Translation of Nonprinting Characters and Control Codes When Extracting an INPUT File from a SESSION File

Code in SESSION File				Translated Special String in INPUT File
Mnemonic	Recording Function	8-bit Control String	Escape Sequence	Special String
—	CTRL/P ¹ D	—	—	{DELAY}
—	CTRL/P ¹ E	—	—	{END_COMPARE}
—	CTRL/P ¹ W	—	—	{WAIT}

¹CTRL/P is used here only as an example. The actual value will be the termination character specified when the SESSION file was recorded.

Table B-3 describes special string translations for control and nonprinting characters when a SESSION file is recorded from an INPUT file. Where applicable special strings for common names of nonprinting characters are listed with the mnemonic for the control character. For example, {BACK_SPACE} is listed with {CTRL/H}.

Table B-3: Translation of Special Strings Representing Control and Nonprinting Characters When Recording a SESSION File from an INPUT File

Special Strings in INPUT File				Translation in SESSION File
Control Character Mnemonics			Decimal Value	
{CTRL/@}	{^@}	{ <NUL> }	{0}	NUL
{CTRL/A}	{^A}	{ <SOH> }	{1}	SOH
{CTRL/B}	{^B}	{ <STX> }	{2}	STX
{CTRL/C}	{^C}	{ <ETX> }	{3}	ETX
{CTRL/D}	{^D}	{ <EOT> }	{4}	EOT
{CTRL/E}	{^E}	{ <ENQ> }	{5}	ENQ
{CTRL/F}	{^F}	{ <ACK> }	{6}	ACK
{CTRL/G}	{^G}	{ <BEL> }	{7}	BEL

Table B-3 (Cont.): Translation of Special Strings Representing Control and Nonprinting Characters When Recording a SESSION File from an INPUT File

Special Strings in INPUT File			Translation in SESSION File	
Control Character Mnemonics		Decimal Value		
{CTRL/H} and {BACK_SPACE}	{^H}	{ <BS> }	{8}	BS
{CTRL/I} and {TAB}	{^I}	{ <HT> }	{9}	HT
{CTRL/J} and {LINE_FEED}	{^J}	{ <LF> }	{10}	LF
{CTRL/K}	{^K}	{ <VT> }	{11}	VT
{CTRL/L} and {PAGE} and {FORM_FEED}	{^L}	{ <FF> }	{12}	FF
{CTRL/M} and {RETURN}	{^M}	{ <CR> }	{13}	CR
{CTRL/N}	{^N}	{ <SO> }	{14}	SO
{CTRL/O}	{^O}	{ <SI> }	{15}	SI
{CTRL/P}	{^P}	{ <DLE> }	{16}	DLE
{CTRL/Q}	{^Q}	{ <DC1> }	{17}	DC1
{CTRL/R}	{^R}	{ <DC2> }	{18}	DC2
{CTRL/S}	{^S}	{ <DC3> }	{19}	DC3
{CTRL/T}	{^T}	{ <DC4> }	{20}	DC4
{CTRL/U}	{^U}	{ <NAK> }	{21}	NAK
{CTRL/V}	{^V}	{ <SYN> }	{22}	SYN
{CTRL/W}	{^W}	{ <ETB> }	{23}	ETB
{CTRL/X}	{^X}	{ <CAN> }	{24}	CAN
{CTRL/Y}	{^Y}	{ }	{25}	EM

Table B-3 (Cont.): Translation of Special Strings Representing Control and Nonprinting Characters When Recording a SESSION File from an INPUT File

Special Strings in INPUT File				Translation in SESSION File
Control Character Mnemonics			Decimal Value	
{CTRL/Z}	{Z}	{ <SUB> }	{26}	SUB
{CTRL/[] {ESCAPE} and {ESC}	{[]}	{ <ESC> }	{27}	ESC
{CTRL/\}	{\}	{ <FS> }	{28}	FS
{CTRL/}	{}	{ <GS> }	{29}	GS
{CTRL/~}	{~}	{ <RS> }	{30}	RS
{CTRL/?} and {CTRL/_}	{?} and {_}	{ <US> }	{31}	US
{DELETE}	—	{ }	{128}	DEL
{SPACE}	—	—	—	(a space character)

Table B-4 describes special string translations for 8-bit control characters when a SESSION file is recorded from an INPUT file. These control characters are available only in 8-bit environments.

Table B-4: Translation of Special Strings Representing 8-Bit Control Characters When Recording a SESSION File from an INPUT File

Special String in INPUT File		Translation in SESSION File
Mnemonic	Decimal Value	
{ <IND> }	{132}	IND
{ <NEL> }	{133}	NEL
{ <SSA> }	{134}	SSA
{ <ESA> }	{135}	ESA
{ <HTS> }	{136}	HTS
{ <HTJ> }	{137}	HTJ
{ <VTS> }	{138}	VTS
{ <PLD> }	{139}	PLD
{ <PLU> }	{140}	PLU
{ <RI> }	{141}	RI
{ <SS2> }	{142}	SS2
{ <SS3> }	{143}	SS3
{ <DCS> }	{144}	DCS
{ <PU1> }	{145}	PU1
{ <PU2> }	{146}	PU2
{ <STS> }	{147}	STS
{ <CCH> }	{148}	CCH
{ <MW> }	{149}	MW
{ <SPA> }	{150}	SPA
{ <EPA> }	{151}	EPA
{ <CSI> }	{155}	CSI

Table B-4 (Cont.): Translation of Special Strings Representing 8-Bit Control Characters When Recording a SESSION File from an INPUT File

Special String in INPUT File		Translation in SESSION File
Mnemonic	Decimal Value	
{ <ST> }	{156}	ST
{ <OSC> }	{157}	OSC
{ <PM> }	{158}	PM
{ <APC> }	{159}	APC

Table B-5 describes special string translations for codes generated by the function keys when a SESSION file is recorded from an INPUT file.

Table B-5: Translation of Special Strings Representing the Function Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File	
	VT200 Mode	VT100 and VT52 Mode
{F6}	CSI 17~ and ESC [¹ 17~	—
{F7}	CSI 18~ and ESC [¹ 18~	—
{F8}	CSI 19~ and ESC [¹ 19~	—
{F9}	CSI 20~ and ESC [¹ 20~	—
{F10}	CSI 21~ and ESC [¹ 21~	—
{F11} and {ESC}	CSI 23~ and ESC [¹ 23~	ESC

¹ESC [is the 7-bit code extension equivalent for the 8-bit control string CSI.

Table B-5 (Cont.): Translation of Special Strings Representing the Function Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File	
	VT200 Mode	VT100 and VT52 Mode
{F12} and {BS}	CSI 24~ and ESC [¹ 24~	BS
{F13} and {LF}	CSI 25~ and ESC [¹ 25~	LF
{F14}	CSI 26~ and ESC [¹ 26~	—
{F15} and {HELP}	CSI 28~ and ESC [¹ 28~	—
{F16} and {DO}	CSI 29~ and ESC [¹ 29~	—
{F17}	CSI 31~ and ESC [¹ 31~	—
{F18}	CSI 32~ and ESC [¹ 32~	—
{F19}	CSI 33~ and ESC [¹ 33~	—
{F20}	CSI 34~ and ESC [¹ 34~	—

¹ESC [is the 7-bit code extension equivalent for the 8-bit control string CSI.

Table B-6 describes special string translations for codes associated with editing keys when a SESSION file is recorded from an INPUT file.

Table B-6: Translation of Special Strings Representing the Editing Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File	
	VT200 Mode	VT100 and VT52 Modes
{FIND}	CSI 1~ and ESC [¹ 1~	—
{INSERT_HERE} and {INSERT}	CSI 2~ and ESC [¹ 2~	—
{REMOVE}	CSI 3~ and ESC [¹ 3~	—
{SELECT}	CSI 4~ and ESC [¹ 4~	—
{PREV_SCREEN} and {PREV}	CSI 5~ and ESC [¹ 5~	—
{NEXT_SCREEN} and {NEXT}	CSI 6~ and ESC [¹ 6~	—

¹ESC [is the 7-bit code extension equivalent for the 8-bit control string CSI.

Table B-7 describes special string translations for codes associated with the keypad keys when a SESSION file is recorded from an INPUT file.

Table B-7: Translation of Special Strings Representing the Keypad Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File			
	ANSI Mode ¹		VT52 Mode ¹	
	Numeric Keypad Mode	Application Keypad Mode	Numeric Keypad Mode	Application Keypad Mode
{KP0}	0	SS3 p and ESC O ² p	0	ESC ? p
{KP1}	1	SS3 q and ESC O ² q	1	ESC ? q
{KP2}	2	SS3 r and ESC O ² r	2	ESC ? r
{KP3}	3	SS3 s and ESC O ² s	3	ESC ? s
{KP4}	4	SS3 t and ESC O ² t	4	ESC ? t
{KP5}	5	SS3 u and ESC O ² u	5	ESC ? u
{KP6}	6	SS3 v and ESC O ² v	6	ESC ? v
{KP7}	7	SS3 w and ESC O ² w	7	ESC ? w
{KP8}	8	SS3 x and ESC O ² x	8	ESC ? x
{KP9}	9	SS3 y and ESC O ² y	9	ESC ? y
{COMMA}	, (comma)	SS3 l and ESC O ² l	, (comma)	ESC ? l

¹ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

²ESC O is the 7-bit code extension equivalent for the 8-bit control string SS3.

Table B-7 (Cont.): Translation of Special Strings Representing the Keypad Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File			
	ANSI Mode ¹		VT52 Mode ¹	
	Numeric Keypad Mode	Application Keypad Mode	Numeric Keypad Mode	Application Keypad Mode
{MINUS}	- (minus)	SS3 m and ESC O ² m	- (minus)	ESC ? m
{PERIOD}	. (period)	SS3 n and ESC O ² n	. (period)	ESC ? n
{ENTER}	CR	SS3 M and ESC O ² M	CR	ESC ? M
{PF1}	SS3 P and ESC O ² P	SS3 P and ESC O ² P	ESC P	ESC P
{PF2}	SS3 Q and ESC O ² Q	SS3 Q and ESC O ² Q	ESC Q	ESC Q
{PF3}	SS3 R and ESC O ² R	SS3 R and ESC O ² R	ESC R	ESC R
{PF4}	SS3 S and ESC O ² S	SS3 S and ESC O ² S	ESC S	ESC S

¹ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

²ESC O is the 7-bit code extension equivalent for the 8-bit control string SS3.

Table B-8 describes special string translations for arrow key codes when a SESSION file is recorded from an INPUT file.

Table B-8: Translation of Special Strings Representing the Arrow Key Codes When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File			
	ANSI Mode ¹		VT52 Mode ¹	
	Cursor Key Mode Reset Normal	Cursor Key Mode Set Application	Cursor Key Mode Reset Normal	Cursor Key Mode Set Application
{UP_ARROW} and {UP}	CSI A and ESC [³ A	SS3 A and ESC O ² A	ESC A	ESC A
{DOWN_ARROW} and {DOWN}	CSI B and ESC [³ B	SS3 B and ESC O ² B	ESC B	ESC B
{RIGHT_ARROW} and {RIGHT}	CSI C and ESC [³ C	SS3 C and ESC O ² C	ESC C	ESC C
{LEFT_ARROW} and {LEFT}	CSI D and ESC [³ D	SS3 D and ESC O ² D	ESC D	ESC D

¹ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

²ESC O is the 7-bit code extension equivalent for the 8-bit control string SS3.

³ESC [is the 7-bit code extension equivalent for the 8-bit control string CSI.

Table B-9 describes special string translations for codes associated with recording functions when a SESSION file is recorded from an INPUT file.

Table B-9: Translation of Special Strings Representing the Recording Functions When Recording a SESSION File from an INPUT File

Special String in INPUT File	Translation in SESSION File
{BEGIN_COMPARE}	DLE ¹ B
{COMPARE_SCREEN}	DLE ¹ C
{END_COMPARE}	DLE ¹ E
{DELAY}	DLE ¹ D
{WAIT}	DLE ¹ W
{BEGIN_COMMENT}	DLE ¹ !
{END_COMMENT}	SUB (CTRL/Z)

¹DLE (CTRL/P) is used here only as an example. The actual value will be the termination character specified when the SESSION file is recorded.



Interactive Testing Example

This appendix shows a complete DEC/Test Manager interactive testing example. The interactive application being used is a VAX Forms Management System (FMS) sample program called SAMP, which does checking account manipulation using Forms Management System. SAMP is installed on the system as part of a standard FMS Version 2.0 installation. FMS Version 2.0 must be installed on your system in order for you to recreate this example.

Before you use DEC/Test Manager, you must create a DEC/Test Manager library in an empty VAX/VMS directory:

```
Ⓢ CREATE/DIRECTORY [PROJECT.DTMLIBRARY]
Ⓢ DTM CREATE LIBRARY [PROJECT.DTMLIBRARY]
  _Remark: PROJECT DEC/TEST MANAGER LIBRARY
%DTM-S-CREATED, DEC/Test Manager library DISK1:[PROJECT.DTMLIBRARY] created
```

After the library is created, you can begin to create and organize a test system. It is necessary to create tests, and then to identify the tests and the test data to DEC/Test Manager. The basic organizational unit within DEC/Test Manager is the **test description**. A test description contains all the information needed to run a particular test and consists of fields which point to the test and to the test-related files. In the following example, a test description is created using the CREATE TEST_DESCRIPTION command. The /RECORD qualifier allows for the capturing of an interactive terminal session and produces a **SESSION** file as the test template. The SESSION file records all input and output.

Create the test description FMSTEST by entering the following CREATE TEST_DESCRIPTION/RECORD command.

⌘ DTM CREATE TEST_DESCRIPTION /RECORD FMSTEST
_Remark: DTM TEST OF FMS CHECKING ACCOUNT EXAMPLE
%DTM-I-DEFAULTED, benchmark file name defaulted to FMSTEST.BMK
%DTM-I-DEFAULTED, template file name defaulted to FMSTEST.SESSION
%DTM-I-BEGIN, your interactive test session is now beginning...
Type CTRL/P twice to terminate the session.

Invoke the sample FMS program.

⌘ RUN FMS\$EXAMPLES:SAMP

Welcome to the FMS V2
Sample Application Program (SAMP)

YOUR PERSONAL CHECKING ACCOUNT

For instructions, press HELP (the PF2 key).
To continue, press RETURN.

ZK-4755-85

Press RETURN to display the main menu.

Checking Account Menu

Choose option (1-5): 2

- 1 Exit
- 2 Write a check
- 3 Make a deposit
- 4 View the check register
- 5 Show account data

For help, press HELP.
To continue, press keypad 1-5.

ZK-4756-85

Select Option 2 to display the check-writing form.

WRITE A CHECK

Katherine M. Smith 1 Hog Hill Rd. Townsend, AK 99999 (800)555-1212	Number <u> 8 </u> Date: <u>22-OCT-85</u> Amount: <u>*****. **</u>
Pay to _____ Memo _____	
FIRST NATIONAL BANK	Account 532

Current Balance: \$ 361.30

ZK-4762-85

Fill in the required check fields. The check is written and sent to the payee's account.

WRITE A CHECK

Katherine M. Smith 1 Hog Hill Rd. Townsend, AK 99999 (800)555-1212	Number <u> 8 </u> Date: <u>22-OCT-85</u>
Pay to <u>DIGITAL EQUIPMENT CORP</u>	Amount: <u>***33.41</u>
Memo _____	
FIRST NATIONAL BANK	Account <u> 532 </u>

Current Balance: \$ 327.89

Your check has been written and sent to the payee's account.
To return to the menu, press keypad period.
To copy check to file SAMPCH.DAT, press keypad zero.
To write another check, press RETURN.

ZK-4757-85

To return to the main menu, press KP. (the keypad period key). Select Option 1 to exit from the FMS sample program.

Checking Account Menu

Choose option (1-5): 1

- 1 Exit
- 2 Write a check
- 3 Make a deposit
- 4 View the check register
- 5 Show account data

For help, press HELP.
To continue, press keypad 1-5.

ZK-4758-85

Terminate the recording session by pressing CTRL/P twice. This saves the output as a benchmark for future comparisons.

⌘
~P

```
%DTM-I-BMK_SAVED, benchmark has been saved in file
DISK1:[PROJECT.DTMLIBRARY]FMSTEST.BMK;1
%DTM-S-RECORDED, test FMSTEST has been successfully recorded
      in file DISK1:[PROJECT]FMSTEST.SESSION
%DTM-S-CREATED, test description FMSTEST created
```

You can now display the test description that was created and examine the contents of the test description fields by entering the SHOW TEST_DESCRIPTION command and specifying the test-name FMSTEST.

⌘ DTM SHOW TEST_DESCRIPTION FMSTEST

Test Descriptions in DEC/Test Manager Library DISK1:[PROJECT.DTMLIBRARY]

```
FMSTEST      "DTM TEST OF FMS CHECKING ACCOUNT EXAMPLE"
  Template   = FMSTEST.SESSION
  Benchmark  = FMSTEST.BMK
  Prologue   = None Specified
  Epilogue   = None Specified
```

When you run FMSTEST to determine whether FMS is still running as you expect it to run, execute FMSTEST and compare the test results with the expected results recorded in the benchmark file FMSTEST.BMK.

Executing tests with DEC/Test Manager is a two-step process:

1. Create a **collection** containing the test or tests you want to execute.
2. Execute the collection either by submitting it for batch execution or by running it interactively on your terminal.

The following command creates the collection FMSTEST containing one test, FMSTEST.

```
§ DTM CREATE COLLECTION FMSTEST FMSTEST
  _Remark: Test of FMS example
%DTM-S-CREATED, collection FMSTEST created
```

Execute the collection interactively using the DTM RUN command.

```
§ DTM RUN FMSTEST
Starting FMSTEST test run...
```

DEC/Test Manager runs the interactive FMS terminal session, stores the test results, and compares test results against the expected output of the test (the benchmark file DEC/Test Manager saved after recording the SESSION file).

After the test has completed running, you can display the collection summary report for collection FMSTEST by entering the SHOW COLLECTION/FULL command. This report summarizes the outcome of your test run.

```
§ DTM SHOW COLLECTION/FULL FMSTEST

Collections in DEC/Test Manager Library DISK1:[PROJECT.DTMLIBRARY]

FMSTEST      1 test    23-OCT-1985 04:08:41 ""
  Command: CREATE COLLECTION FMSTEST FMSTEST "test FMS example"
  Status: has been run, compared, not reviewed
  Successful count: 0      Unsuccessful count: 1
  New test count: 0      Updated test count: 0
  Test not run count: 0   Comparisons aborted: 0
  Default template directory: SYS$DISK: []
  Template class: 1+
  Default benchmark directory: DTM$LIB:
  Benchmark class: 1+
  Prologue: None Specified
  Epilogue: None Specified
  Result Description FMSTEST      Comparison: Unsuccessful
```



```
Benchmark: DISK1:[PROJECT.DTMLIBRARY]FMSTEST.BMK
Result file is present
Difference file is present
```

Because the output of the current FMSTEST run did not compare successfully with the expected test output, use the DEC/Test Manager Review subsystem to examine the test results. Enter the DTM REVIEW command for collection FMSTEST to invoke the Review subsystem and to examine the results of the collection run.

```
‡ DTM REVIEW FMSTEST
```

```
Collection FMSTEST with 1 test was created on 23-OCT-1985 04:08:41 by the
command:
```

```
CREATE COLLECTION FMSTEST FMSTEST "test FMS example"
Last Review Status = not previously reviewed
Success count = 0
Unsuccessful count = 1
New test count = 0
Updated test count = 0
Comparisons aborted = 0
Test not run count = 0
```

```
DTM_REVIEW>
```

Enter the Review subsystem SELECT command to select test results for FMSTEST, the only test in the collection.

```
DTM_REVIEW> SELECT FMSTEST
Result Description FMSTEST          Comparison Status : Unsuccessful
```

Enter the SHOW/DIFFERENCES command to display the differences between the actual test results and the expected test results.

```
DTM_REVIEW> SHOW/DIFFERENCES
```

Interactive Compare
Version 2.0
Type PF2 for help.

Test	FMSTEST
Result File	DISK1:[PROJECT.DTMLIBRARY.FMSTEST]FMSTE

Interactive Compare
Version 2.0
Type CTRL/Z to exit.

Test	FMSTEST
Benchmark File	DISK1:[PROJECT.DTMLIBRARY]FMSTEST.BMK

Res
Screen 0

Bnk
Screen 0

ZK-4759-85

Press KP0 to move through the differences screens.

WRITE A CHECK

Res
Screen 10

Katherine M. Smith 1 Hog Hill Rd. Townsend, AK 99999 (800)555-1212	Number 8 Date: 28-OCT-85
Pay to DIGITAL EQUIPMENT CORP	Amount: ***33.41
Memo	

WRITE A CHECK

Bmk
Screen 10

Katherine M. Smith 1 Hog Hill Rd. Townsend, AK 99999 (800)555-1212	Number 8 Date: 22-OCT-85
Pay to DIGITAL EQUIPMENT CORP	Amount: ***33.41
Memo	

ZK-4760-85

When the test ran, the only difference produced was the date stamp on the check.

Enter the Review subsystem EXIT command to leave the Review subsystem.

```
DTM_REVIEW> EXIT
%DTM-S-EXIT, leaving Review subsystem
$
```



Glossary

Benchmark directory A VAX/VMS directory or DEC/CMS library used by default for benchmark files in the current DEC/Test Manager library.

Benchmark file A file containing the expected output of a test. A benchmark file can be created during a Review session with the UPDATE command. A benchmark file for an interactive test also can be created when the interactive terminal session is recorded. You can also create benchmark files manually, but doing so is not recommended.

Collection One or more tests selected for execution as a set.

Collection command file A command file created by DEC/Test Manager to execute a collection.

Collection epilogue file An optional command file you create, such as a cleanup file, that runs after all tests in the collection have executed.

Collection-expression A generalized command parameter specifying one or more collections.

Collection-name A command parameter or name specifying a particular collection.

Collection prologue file An optional command file you create, such as a setup file, that runs just before the first test in the collection executes.

Collection subdirectory A subdirectory created by DEC/Test Manager of the library where collection-specific files are stored.

Collection summary A brief summary of the results of running a collection displayed when you enter the Review subsystem and when you enter the Review subsystem SHOW/SUMMARY command.

Current A term used to designate the DEC/Test Manager entity you are currently using or accessing. For example, the current library is the DEC/Test Manager library you are currently using.

DEC/CMS Code Management System. A software library system that stores files and records changes made to and user access of those files.

DEC/CMS class A set of element generations with only one generation per element.

DEC/CMS element An ASCII file stored in a DEC/CMS library.

DEC/CMS generation Representation of a phase in the development of an element—every time you retrieve and then return an element to the library a new generation is created. Any generation of an element can be retrieved; each generation reflects the changes that were made at that particular point in development.

DEC/CMS library The largest group of files that DEC/CMS organizes.

DEC/CMS line of descent A series of generations of an element, created by successive reservation and replacement transactions.

DEC/CMS variant line of descent A line of descent separate from the main line of descent—an alternate development path. Generation numbers of variant line generations consist of combinations of numbers and variant letters.

Difference file A file listing the differences between the result file and benchmark file for a test. DEC/Test Manager generates a difference file only if the result file and benchmark file do not match.

Directory-spec The directory specification portion of a legal VAX/VMS file specification.

DTM\$COLLECTION_NAME A global string symbol defined by DEC/Test Manager which points to a collection. This variable is defined only while the collection is executing.

DTM\$INIT A user-defined VAX/VMS logical name pointing to a DEC/Test Manager initialization file.

DTM\$LIB A logical name defined by DEC/Test Manager which points to the current DEC/Test Manager library.

DTM\$RESULT A logical name defined by DEC/Test Manager which points to the result file for a specific test. This variable exists only while the test epilogue is executing.

DTM\$TEST_NAME A local VAX/VMS symbol defined by DEC/Test Manager which points to a test. This variable exists only while that test is executing.

Epilogue file See **collection epilogue file** or **test epilogue file**.

Expression A generalized command parameter specifying multiple instances of one or more parameters in a single parameter field. An expression can consist of a name, a wildcard character, a wildcard character used in conjunction with a name or partial name, or a list of these separated by commas. The only exception is the result-description-expression which cannot consist of a list separated by commas.

Field A test description entity used to associate specific information with a test description. The test description fields are test-name, template, benchmark, test prologue, test epilogue, variables, groups, filters, and remark. See also **test description** and **field value**.

Field value A value for a test description field. When you supply a value for a test description field, you associate specific files, variables, filters, or other attributes with the associated test. See also **test description** and **field**.

File-name The file-name portion of a legal VAX/VMS file specification.

File-spec A legal VAX/VMS file specification. See the *VAX/VMS DCL Dictionary* for more information.

Filter One of six files provided by DEC/Test Manager to remove run-time variable information from your result file.

Group A named set of test descriptions (usually having common characteristics) that can be manipulated as a unit.

Group-expression A generalized command parameter specifying one or more groups.

Group-name A command parameter or name specifying a particular group of test descriptions.

History A historical record of commands that change the DEC/Test Manager library. For example, CREATE and DELETE commands are logged; SHOW commands are not logged.

INPUT file An editable file containing a text representation of all user input, nonprinting control characters, and recording functions in a SESSION file. Computer output is not contained in a SESSION file.

Initialization file A DEC/Test Manager command file to be executed whenever DEC/Test Manager is invoked as a subsystem.

Interactive test A test whose template is a SESSION file. See also **SESSION file**.

Keypad key The control, arrow, and keypad keys. These keys are user-definable using the DTM DEFINE/KEY command.

Library A VAX/VMS directory containing the DEC/Test Manager control file, result files, difference files, DEC/Test Manager collection subdirectories, and optionally benchmark and SESSION files.

Noninteractive test A test whose template is a command file.

Object-expression A generalized command parameter specifying one or more tests, groups of tests, collections of tests, or variables. Object-expressions encompass all types of DEC/Test Manager entities.

Parameter qualifier A qualifier (either /GROUP or /TEST_DESCRIPTION) used after each item in a test-group-expression to identify the item as either a test-name or a group-name.

Prologue file See **collection prologue file** or **test prologue file**.

Regression testing A testing method which ensures that software being developed or modified runs correctly. As new features are added, the software is repeatedly tested to verify that new features do not affect the correct execution of the previously tested features. When errors exist, the software is said to have "regressed".

Remark A comment associated with a DEC/Test Manager command and recorded in the DEC/Test Manager history file.

Result description A summary of test results accessible from the Review subsystem. DEC/Test Manager generates a result description for every test description in a collection.

Result-description-name A command parameter or name specifying a particular result description. The result-description-name for a test is the same as its test-name.

Result-description-expression A generalized command parameter specifying one or more result descriptions. A result-description-expression cannot consist of a list separated by commas. It can consist of a result-description-name, a wildcard character, or a wildcard character used in conjunction with a full or partial result-description-name.

Result file A file containing the results of a test's execution and accessible only from within the Review subsystem.

Review subsystem The DEC/Test Manager subsystem that allows you to examine and manipulate the results of running a collection of tests.

SESSION file A file containing a recording of an interactive terminal session.

Template directory A directory or DEC/CMS library used by default for template files in the current DEC/Test Manager library.

Template file A VAX/VMS command procedure that executes a noninteractive test, or a SESSION file containing a recorded interactive terminal session.

Termination character The character which, when entered twice, terminates the recording of an interactive terminal session. The default termination character is CTRL/P.

Test description A collection of fields for which you supply values which point to the files and other entities associated with the test. A test description contains all the information DEC/Test Manager needs to run a particular test. Each test must have a corresponding test description. See also **field** and **field value**.

Test epilogue file An optional command file you specify, such as a filter file, that is associated with a test and runs just after the test executes.

Test-expression A generalized command parameter specifying one or more test descriptions.

Test-group-expression A generalized command parameter specifying one or more test descriptions or groups. The parameter qualifiers /GROUP and /TEST_DESCRIPTION identify the individual items in a test-group-expression as specifying either groups or test descriptions.

Test-name A command parameter or name specifying a particular test description.

Test prologue file An optional command file, such as a setup file, that is associated with a test and runs just before the specified test.

Variable A DCL symbol or logical name that you define in a DEC/Test Manager library and associate with collections or tests (or both) in that library.

Variable-expression A generalized command parameter specifying one or more variables.

Variable-name A command parameter or name specifying a particular variable.

Variable-value A value assigned to a particular variable.

VAX Performance and Coverage Analyzer A software development tool that collects performance and coverage data on a program and allows you to interactively analyze that data. When DEC/Test Manager is used with the VAX Performance and Coverage Analyzer, DEC/Test Manager invokes the Collector of the VAX Performance and Coverage Analyzer to gather performance and coverage data while tests are running. After the tests have run, you invoke the Analyzer of the VAX Performance and Coverage Analyzer from the DEC/Test Manager Review subsystem to examine the data gathered.

INDEX

(\$) DTM • 1-21 table, CD-15
/NOINIT • 1-21 table, CD-15
(%) Percent sign • CD-8
(*) Asterisk • CD-8
@file-spec • CD-17
@file-spec (Review subsystem command) • CD-116
{ and } Braces in SESSION file • CD-74

A

Abbreviating commands
See Commands, abbreviating
Access control list
See ACL
Access control list entry
See ACE
ACE (access control list entry) • 2-5
ACL (access control list) • 2-5
defining group access • 2-5
Analyzer Filter
See Performance and Coverage Analyzer,
Analyzer filter
Asterisk (*) • CD-8
ATTACH • 6-35, CD-18
/IDENTIFICATION • CD-18
/PARENT • CD-18
process-name • CD-18
ATTACH (Review subsystem command) • CD-117
/IDENTIFICATION • CD-117
/PARENT • CD-117
process-name • CD-117
Attaching to a process • CD-18, CD-118

Automatic screen compare
See Compare, automatic

B

BACK (Review subsystem command) • 5-8,
CD-119
/COMPARISON_ABORTED • CD-119
comparison status qualifiers • 5-8
count • CD-120
/NEW • CD-119
/NOT_RUN • CD-119
/SUCCESSFUL • CD-119
/UNSUCCESSFUL • CD-119
/UPDATED • CD-120
Batch and interactive environments
effects on collection execution • 4-9
Benchmark directory
canceling the default • 3-27, CD-157
displaying the default • 4-12, CD-167
establishing a hierarchy • 3-26
establishing the default • 3-26, CD-157
overriding the default • 3-26, CD-38, CD-86,
CD-157
Benchmark file • 1-8, 3-2 table, 3-3, 4-2, 4-10,
5-1
See also Benchmark directory
associating a file-spec with • CD-86
BMK_SCREEN file
See Benchmark file, printable screens file
comparing with the result file • 4-2, 4-10
creating • 4-2, 5-18, CD-151
creating from a result file • 4-2
default file-name • 3-2, 3-22, CD-44

Benchmark file (cont'd.)

- displaying
 - for an interactive test • 3-17
 - for interactive tests • CD-70
 - in the Review subsystem • CD-140
- displaying for an interactive test • 5-11
- examining from the Review subsystem • 5-1
- modifying
 - See Benchmark file, updating
- printable screens file (test-name.BMK_
SCREENS) • 5-3 list, 5-21
- removing a file-spec from • CD-86
- replacing • 3-20, CD-86, CD-152
- replacing with the default • 3-20
- saving multiple versions • CD-157
- storing • 3-3, 3-26
 - in a CMS library • 3-27
 - in the DEC/Test Manager library • 3-3
- updating • 5-18
 - from a new result file • 4-2
 - in the DEC/Test Manager library • 5-18
 - outside the DEC/Test Manager library • 5-18
 - with an existing benchmark file • 5-18
 - with no existing benchmark file • 5-19

BMK_SCREEN file

- See Benchmark file, printable screens file

Braces { and } in SESSION file • CD-74

C

Canceling commands

- See Commands, canceling

Checking a library for errors

- See Library, verifying

CMS

- class • 3-27, CD-32, CD-157, CD-165, CD-168
- commands • 5-19, CD-157, CD-165
 - CMS CREATE ELEMENT • 3-28, 3-29, CD-151
 - CMS FETCH • 3-28, 3-30, CD-138
 - CMS INSERT • CD-157, CD-165
 - CMS INSERT/SUPERSEDE • 3-28, CD-151
 - CMS REPLACE • 3-28, 3-29, CD-151, CD-157, CD-165

CMS

commands (cont'd.)

- CMS RESERVE • 3-28, 3-29, CD-151, CD-157, CD-165
 - INSERT/SUPERSEDE • CD-32, CD-157, CD-165, CD-168
 - issued by DEC/Test Manager • 3-27
 - element • 3-27
 - generation • 3-27, 3-30
 - library • 2-3, 3-26
 - creating • 3-27
 - storing benchmark files in • 3-27
 - storing template files in • 3-27
 - variant line of descent • 3-28, CD-151
 - CMS CREATE ELEMENT • 3-28, 3-29, CD-151
 - CMS FETCH • 3-28, 3-30, CD-138
 - CMS INSERT • CD-157, CD-165
 - CMS INSERT/SUPERSEDE • 3-28, CD-151, CD-157, CD-165
 - CMS REPLACE • 3-28, 3-29, CD-151, CD-157, CD-165
 - CMS RESERVE • 3-28, 3-29, CD-151, CD-157, CD-165
- #### Code Management System
- See CMS
- #### Collection epilogue file
- See Epilogue file, collection
- #### Collection-expression • CD-6 list, CD-7, CD-100
- #### Collection-name • 1-15, 4-3, CD-2, CD-22
- restrictions on • CD-2
- #### Collection prologue file
- See Prologue file, collection
- #### Collections • 1-15, 4-1
- aborted
 - See Collections, improperly stopped
 - as a snapshot of tests • 4-3
 - command file • 4-1, 4-5
 - comparing • 4-10
 - automatically • 4-11
 - manually • 4-11
 - when partially run • 4-10
 - containing an incomplete test • 4-4
 - creating • 1-15, 4-1, 4-3, 4-5, CD-37
 - example • 4-4, C-6
 - including a group of tests • 4-5
 - resolution of file specifications • 4-4
 - verification of collection prologue file • 4-4

Collections

- creating (cont'd.)
 - with missing files • 4-4
 - without automatic execution • 4-5
 - without resolving file specifications • 4-4
- deleting • 4-13, CD-60
 - example • 4-14
 - problems caused by file protections • 4-14
 - when to delete • 4-14
- description of • 4-3
- displaying information about • CD-170
 - example • 4-12
- displaying summary information • 4-12
 - example • C-6
- executing • 1-15, 4-1, 4-5, 4-6 figure
/COMPARE • 4-4
 - effects of differing batch and interactive environments • 4-9
 - example • 4-8, 4-9, C-6
 - in batch • 1-16, 4-6, 4-7, CD-37, CD-155, CD-199
 - interactively • 1-16, 4-6, 4-9, CD-37, CD-154
 - large • 4-9, CD-154
 - missing test prologue file • 4-5
 - missing test template file • 4-5
 - /NOCOMPARE • 4-4
 - /SUBMIT • 4-4
 - with the RUN command • 4-4
 - with the SUBMIT command • 4-4
- for use with the Performance and Coverage Analyzer • 7-11
 - See also Performance and Coverage Analyzer
 - command file • 7-6
 - creating • 7-5
 - executing • 7-6
 - prologue file • 7-4
- improperly stopped
 - recovering from • 6-32
- incorporating changed files • 4-14
- incorporating changes to library • 4-14
- modifying • CD-37
- partially run • 4-5, 4-10
 - causes • 5-23
 - comparing • 4-10, CD-23
 - executing remaining tests • 4-10

Collections

- partially run (cont'd.)
 - reviewing • 5-23, 5-24
 - recreating • 4-3, 4-14, CD-37, CD-100, CD-199 list
 - process for • 4-14
 - restriction on • 4-14
 - to incorporate a new global variable values • 6-6
 - when to recreate • 4-14
- reexecuting • 4-9
- requirements for creation • CD-37
- rerunning • CD-37, CD-199 list
- resubmitting • 4-8
- reviewing • CD-109
- stopping
 - improperly • 5-23, 5-24
- stopping execution of • 4-15, CD-155, CD-197
 - errors • 4-15
 - example • 4-15
 - when running in batch • 4-15, 5-24
 - when running interactively • 4-15, 5-24
 - with CTRL/C • 4-15, 5-24
 - with the STOP command • 4-15, 5-24
- summary • 4-13
 - displaying • 4-12
 - when to modify • 4-3
 - when to recreate • 4-3
- Collection Summary Information
 - possible errors for read-only reviewers • 5-6
- Command file
 - description of • 6-32
 - entering commands in • 6-33
 - entering comments in • 6-33
 - executing • 6-32, CD-17, CD-116
 - EXIT command in • 6-33
 - invoking • 6-32
 - invoking DEC/Test Manager from • 6-33
 - nesting • 6-33
 - restriction on the use of hyphen • 6-33
 - terminating • 6-33
 - writing • 6-33
- Command parameter
 - See Parameter
- Command procedure
 - See Command file

Command qualifier

See Qualifier, command

Commands

abbreviating • CD-11
access required • 2-6 table
by function • 1-20 to 1-28 tables
canceling • CD-11
entering • 1-6, CD-10
for creating a library • 1-21 table
for getting started • 1-21 table
format of • CD-1
for reviewing test results • 1-26 table
for using a library • 1-22 table
general form of • CD-1
name • CD-1
syntax of • CD-1 to CD-11

Commands, DTM

ATTACH • CD-18 to CD-19
COMPARE • 1-26 table, CD-20 to CD-24
CONVERT LIBRARY • 1-21 table, 2-9, 2-10,
CD-25 to CD-26
COPY TEST_DESCRIPTION • 1-23 table, CD-27
to CD-30
CREATE COLLECTION • 1-15, 1-26 table,
CD-31 to CD-38
CREATE GROUP • 1-24 table, CD-39 to CD-40
CREATE LIBRARY • 1-7, 1-21 table, 2-9,
CD-41 to CD-42
CREATE TEST_DESCRIPTION • 1-23 table,
CD-43 to CD-50
/INPUT • 1-23 table
/RECORD • 1-23 table
CREATE TEST_DESCRIPTION/RECORD • 1-8,
1-12
CREATE VARIABLE • 1-24 table, CD-51 to
CD-54
CTRL/C
See Commands, canceling • CD-11
CTRL/Z
See also EXIT
See also Exiting, DEC/Test Manager
subsystem
DEFINE/KEY • CD-55 to CD-58
DELETE COLLECTION • 1-26 table, CD-59 to
CD-60
DELETE GROUP • 1-24 table, CD-61 to CD-62
DELETE HISTORY • CD-63 to CD-64

Commands, DTM (cont'd.)

DELETE TEST_DESCRIPTION • 1-23 table,
CD-65 to CD-67
DELETE VARIABLE • 1-24 table, CD-68 to
CD-69
DISPLAY • 1-23 table, 3-17, CD-70 to CD-71
(\$) DTM • CD-15 to CD-16
EXIT • 1-21 table, CD-11, CD-72
EXTRACT • 1-23 table, CD-73 to CD-75
@file-spec • CD-17
HELP • 1-4, 1-21 table, CD-76
INSERT GROUP • 1-24 table, CD-77 to CD-79
INSERT TEST_DESCRIPTION • 1-24 table,
CD-80 to CD-82
MODIFY GROUP • 1-24 table, CD-83 to CD-84
MODIFY TEST_DESCRIPTION • 1-23 table,
CD-85 to CD-93
/NOPROLOGUE • 3-29
MODIFY VARIABLE • 1-24 table, CD-94 to
CD-96
PLAY • 1-23 table, CD-97 to CD-98
RECREATE • 1-26 table, CD-99 to CD-100
REMARK • CD-101 to CD-102
REMOVE GROUP • 1-24 table, CD-103 to
CD-104
REMOVE TEST_DESCRIPTION • 1-24 table,
CD-105 to CD-107
REVIEW • 1-18, CD-108 to CD-152
RUN • 1-16, 1-26 table, CD-153 to CD-155
SET BENCHMARK_DIRECTORY • 1-25 table,
CD-156 to CD-157
SET EPILOGUE • 1-25 table, CD-158 to
CD-159
SET LIBRARY • 1-7, 1-21 table, 2-10, CD-160
to CD-161
/NOVERIFY • 2-10
SET NOBENCHMARK_DIRECTORY • CD-156,
CD-157
SET NOEPILOGUE • CD-159
SET NOPROLOGUE • CD-163
SET NOTEMPLATE_DIRECTORY • CD-165
SET PROLOGUE • 1-25 table, CD-162 to
CD-163
SET TEMPLATE_DIRECTORY • 1-25 table,
CD-164 to CD-165
SHOW ALL • 1-25 table, CD-166
SHOW BENCHMARK_DIRECTORY • 1-25 table,
CD-167

Commands, DTM (cont'd.)

SHOW COLLECTION • 1-26 table, **CD-168** to **CD-170**
SHOW EPILOGUE • 1-25 table, **CD-171**
SHOW GROUP • 1-24 table, **CD-172** to **CD-174**
SHOW HISTORY • **CD-175** to **CD-179**
SHOW LIBRARY • 1-21 table, 2-10, **CD-180**
SHOW PROLOGUE • 1-25 table, **CD-181**
SHOW TEMPLATE_DIRECTORY • 1-25 table, **CD-182**
SHOW TEST_DESCRIPTION • 1-23 table, **CD-183** to **CD-187**
SHOW VARIABLE • 1-24 table, **CD-188** to **CD-190**
SHOW VERSION • 1-21 table, **CD-191**
SPAWN • **CD-192** to **CD-195**
STOP • 1-26 table, **CD-196** to **CD-197**
SUBMIT • 1-16, 1-26 table, **CD-198** to **CD-200**
/NOTIFY • 1-16
VERIFY • **CD-201** to **CD-202**

Commands, DTM_REVIEW

ATTACH • 1-27 table
BACK • 1-28 table
CTRL/C • 1-27 table
CTRL/Z • 1-27 table
DEFINE/KEY • 1-27 table
EXIT • 1-27 table
EXIT/NOPRINT/NOINSERT • 1-27 table
@file-spec • 1-27 table
FIRST • 1-28 table
HELP • 1-4, 1-27 table
INSERT • 1-28 table
LAST • 1-28 table
NEXT • 1-28 table
PCA • 1-27 table
PRINT • 1-28 table
SELECT • 1-28 table
SHOW • 1-28 table
SHOW/FILES • 1-28 table
SHOW/SUMMARY • 1-28 table
SPAWN • 1-27 table
UPDATE • 1-28 table

Comment record • B-4

COMPARE • 1-26 table, 4-4, 4-11, CD-20
/CHARACTERS • CD-20
/FULL • CD-20

COMPARE (cont'd.)

/IGNORE • CD-21
/LOG • CD-21
/NOLOG • CD-21
/NOPARALLEL • CD-21
/PARALLEL • CD-21
/RECORDS • CD-22
/SCREENS • CD-22
/WIDTH • CD-22

Compare

automatic • 3-9, 3-11, CD-23
screens included in benchmark file • 3-11
screens marked for comparison • 3-11
selecting • 3-11, 3-21
terminating • 3-11
manual • CD-23
marking screens for comparison • 3-11
marking screens for inclusion in the benchmark file • 3-11
selecting • 3-11, 3-21
terminating • 3-11
verifying the selected screens • 3-17
specifying comparison type • 3-21

Comparing

interactive tests
character by character • 4-11, CD-23
record by record • 4-11, CD-22, CD-24
screen by screen • 4-11, CD-23
partially run collections • CD-23

Comparison status • 4-2, 4-10, 4-11, CD-23

comparison aborted • 4-11, 5-4, CD-24
description of • 5-4
information provided by • 5-4
new test • 4-11, 5-4, CD-23
not run • 4-11, 5-4, CD-23
successful • 4-11, 5-4, CD-23
unsuccessful • 4-11, 5-4, CD-23
updated • 4-11, 5-4, CD-23

Comparison status qualifiers • 5-4, 5-9, CD-110, CD-111, CD-138, CD-144
/COMPARISON_ABORTED • CD-138, CD-144
/NEW • CD-138, CD-144
/NOT_RUN • CD-138, CD-144
on the PRINT command • CD-138
on the SHOW command • CD-144
/SUCCESSFUL • CD-138, CD-144
/UNSUCCESSFUL • CD-138, CD-144
/UPDATED • CD-138, CD-144

/CONFIRM • CD-9
 Confirmation • CD-9
 Converting a library
 See Library, converting to the current version
 CONVERT LIBRARY • 1-21 table, 2-9, 2-10, CD-25
 COPY TEST_DESCRIPTION • 1-23 table, 3-22,
 CD-27
 /COMPARISON_TYPE • CD-27
 /EPILOGUE • CD-28
 /LOG • CD-28
 /NOEPILOGUE • CD-28
 /NOFILTERS • 6-36, CD-28
 /NOGROUPS • CD-28
 /NOLOG • CD-28
 /NOPROLOGUE • CD-28
 /NOREMARK • CD-29
 /NOVARIABLES • CD-29
 /PROLOGUE • CD-28
 /REMARK • CD-29
 remark • CD-30
 remark parameter and remark qualifier • CD-4
 /TEMPLATE • CD-29
 test-name • CD-29, CD-30
 CREATE COLLECTION • 1-26 table, 4-3, CD-31
 /BENCHMARK_DIRECTORY • CD-31
 /CLASS • CD-32
 collection-name • CD-36
 /COMPARE • CD-32
 /EPILOGUE • 3-29, CD-33
 example • C-6
 /GROUP • CD-36
 /LOG • CD-33
 /NOBENCHMARK_DIRECTORY • CD-32
 /NOCOMPARE • CD-23, CD-33
 /NOEPILOGUE • 3-29, CD-33
 /NOLOG • CD-33
 /NOPROLOGUE • 3-29, CD-34
 /NOSUBMIT • 4-8, CD-34
 /NOTEMPLATE_DIRECTORY • CD-35
 /NOVERIFY • 4-4, CD-35
 /PROLOGUE • 3-29, CD-34
 remark • CD-36
 /SUBMIT • 4-7, CD-34, CD-37
 /TEMPLATE_DIRECTORY • CD-35
 /TEST_DESCRIPTION • CD-36
 test-group-expression • CD-36
 /VARIABLE • CD-35

CREATE COLLECTION (cont'd.)
 /VERIFY • CD-35
 CREATE GROUP • 1-24 table, 6-15, CD-39
 group-name • CD-39
 /LOG • CD-39
 /NOLOG • CD-39
 remark • CD-39
 CREATE LIBRARY • 1-21 table, 2-9, CD-41
 directory-spec • CD-41
 /LOG • CD-41
 /NOLOG • CD-41
 remark • CD-41
 CREATE TEST_DESCRIPTION • 1-23 table, 3-1,
 3-6, CD-43
 /BENCHMARK • CD-43
 /EPILOGUE • 3-29, CD-44
 /FILTER • 6-35 table, CD-44
 /INPUT • 1-23 table, CD-45, CD-74
 /INTERACTIVE • CD-46
 /LOG • CD-46
 /NOINTERACTIVE • CD-46
 /NOLOG • CD-46
 /PROLOGUE • 3-29, CD-46
 /RECORD • 1-23 table, CD-46
 example • C-1
 remark • CD-48
 /TEMPLATE • CD-47
 /TERMINATION_CHARACTER • CD-47
 test-name • CD-48
 /VARIABLE • CD-48
 CREATE TEST_DESCRIPTION/RECORD • 3-8
 CREATE VARIABLE • 1-24 table, 6-4, CD-51,
 CD-91
 /GLOBAL • CD-51
 LOCAL • CD-51
 /LOG • CD-51
 /LOGICAL • CD-52
 /NOLOG • CD-51
 /NUMERIC • CD-52
 remark • CD-53
 /STRING • CD-52
 /SYMBOL • CD-52
 variable-name • CD-52
 variable-value • CD-53
 Creating a library
 See Library, creating

Creating collections

See Collections, creating

Creating groups

See Groups, creating

Creating test descriptions

See Test description, creating

Creating variables

See Variables, creating

CTRL/A • B-10

double in SESSION file • B-10

single in SESSION file • B-10

CTRL/C • 1-27 table, 4-15, 5-23, 5-24, 5-25, CD-11, CD-23, CD-155, CD-197

See also Collections, aborting execution of

See also Commands, canceling

CTRL/P • 1-14, 3-10, CD-73

See also Recording functions

See also Termination character, default

CTRL/Y • 5-23, CD-155

restriction on use • 3-13, CD-155

CTRL/Z • 1-21 table, 1-27 table, 5-24, CD-11, CD-72, CD-125

See also Exiting, DEC/Test Manager subsystem

See also Exiting, Review subsystem

D

DCL

command file • CD-11

continuation character • CD-10

SUBMIT command • 4-8

the DELETE/ENTRY command • 5-23

using the DCL SUBMIT command qualifiers •
4-8, CD-199 list

DEC/CMS

See CMS

DEC/Test Manager

description of • 1-1

exiting subsystem • CD-11, CD-72

exiting the subsystem • 1-6

invoking as a subsystem • 1-21, CD-15

subsystem prompt • 1-5

terminating a session • 1-6

DEC/Test Manager prompt

See DTM>

DEFINE/KEY • CD-55

command-string • CD-57

/ECHO • CD-55

/IF_STATE • CD-55

key-name • CD-56

key names • CD-56-Table

/LOCK_STATE • CD-55

/NOECHO • CD-55

/NOIF_STATE • CD-55

/NOLOCKSTATE • CD-55

/NOSET_STATE • CD-56

/NOTERMINATE • CD-56

/SET_STATE • CD-56

/TERMINATE • CD-56

DEFINE/KEY (Review subsystem command) • 5-27, CD-121

command-string • CD-123

/ECHO • CD-121

/IF_STATE • CD-121

key-name • CD-122

/LOCK_STATE • CD-121

/NOECHO • CD-121

/NOIF_STATE • CD-121

/NOLOCK_STATE • CD-121

/NOSET_STATE • CD-122

/NOTERMINATE • CD-122

/SET_STATE • CD-122

state-name • CD-123

/TERMINATE • CD-122

Defining keypad keys

DEC/Test Manager subsystem • CD-57

Review subsystem • CD-123

DELETE COLLECTION • 1-26 table, 4-13, CD-59

collection-expression • CD-59

/CONFIRM • CD-59

/LOG • CD-59

/NOCONFIRM • CD-59

/NOLOG • CD-59

remark • CD-60

DELETE GROUP • 1-24 table, 6-22, CD-61

/CONFIRM • CD-61

group-expression • CD-61

/LOG • CD-61

/NOCONFIRM • CD-61

/NOLOG • CD-61

remark • CD-61

DELETE HISTORY • CD-63
 /BEFORE • CD-63
 /CONFIRM • CD-63
 /LOG • CD-63
 /NOCONFIRM • CD-63
 /NOLOG • CD-63
 /OUTPUT • CD-64
 remark • CD-64
 DELETE TEST_DESCRIPTION • 1-23 table, 3-25,
 CD-65
 /CONFIRM • CD-65
 /LOG • CD-65
 /NOCONFIRM • CD-65
 /NOLOG • CD-65
 remark • CD-65
 test-expression • CD-65
 DELETE VARIABLE • 1-24 table, 6-6, CD-68
 /CONFIRM • CD-68
 /LOG • CD-68
 /NOCONFIRM • CD-68
 /NOLOG • CD-68
 remark • CD-68
 variable-expression • CD-68
 Deleting collections
 See Collections, deleting
 Deleting groups
 See Groups, deleting
 Deleting history information
 See History, deleting
 Deleting test descriptions
 See Test description, deleting
 Deleting variables
 See Variables, deleting
 Diagnostic messages
 See Messages
 Difference file • 1-16, 4-2, 5-1, CD-23
 deleting • CD-66
 description of • 5-3 list
 displaying • CD-141
 displaying all screens • 5-15
 displaying for an interactive test • 5-10, 5-13
 displaying only screens with differences • 5-15
 examining from the Review subsystem • 5-1
 highlighting of differences • 5-13, 5-15
 changing • 5-13
 restriction on changing • 5-14 table
 split-screen mode • 5-15

Directory-spec parameter • CD-3
 DISPLAY • 1-23 table, CD-70
 /BENCHMARK • 3-17, CD-70
 test-name • CD-70
 Displaying the history file
 See History file, displaying
 DTM
 /NOINIT • 6-34
 See also (\$) DTM
 DTM\$\$TEST_RUN.COM • 2-4
 DTM\$COLLECTION_NAME • 4-5, 6-2, 6-12, 6-13
 description of • 6-13
 example • 6-13
 uses of • 6-13
 when available • 6-13
 DTM\$INIT • 6-33, 6-34
 DTM\$LIB • 2-9, 2-10, 3-3, 3-22
 DTM\$RESULT • 4-5, 6-2, 6-11, CD-22
 description of • 6-11
 uses of • 6-11
 when defined • 6-11
 DTM\$TEST_NAME • 4-5, 6-12
 description of • 6-12
 example • 6-12
 uses of • 6-12
 DTM> (subsystem prompt) • 1-5
 DTM_REVIEW
 PCA • 7-9
 DTM_REVIEW> (Review subsystem prompt) •
 CD-108
 DTM_REVIEW commands
 See Commands, DTM_REVIEW

E

Entering commands
 See Commands, entering
 Epilogue file
 collection • 1-15, 6-3
 canceling the default • 3-29, 6-3, CD-159
 description of • 6-2
 displaying the default • CD-171
 establishing the default • 3-29, 6-3,
 CD-159
 overriding the default • 3-29, 6-3, CD-159
 storing • 3-26
 storing in a CMS library • 3-30

Epilogue file

- collection (cont'd.)
 - when executed • 6-2
- description of • 6-1
- storing • 2-3, 3-30
- test • 1-8, 3-2 table, 3-4, 6-2
 - adding to a test description • 3-29, 6-2, CD-87
 - canceling • 3-29
 - canceling for a test description • 3-20, CD-87
 - description of • 3-4, 6-2
 - determining the need for • 6-2
 - removing from a test description • 3-20, 6-3, CD-87
 - replacing for a test description • 3-20, 3-29, CD-87
 - storing • 3-26
 - storing in a CMS library • 3-30
 - uses of • 6-2
 - when executed • 6-2
- using • 6-1
- using DTM\$RESULT in • 6-11
 - example • 6-11
- Establishing a DEC/Test Manager library
 - See Library, creating
- EXIT • 1-21 table, 1-27 table, CD-11, CD-72
 - (CTRL/Z) • 1-27 table
- EXIT (Review subsystem command) • 5-24, CD-125
 - example • C-9
 - /INSERT • 5-25
 - /NOINSERT • 5-25, CD-125
 - /NOPRINT • 5-25, CD-125
 - /PRINT • 5-25
- EXIT/NOPRINT/NOINSERT • 1-27 table
 - (CTRL/C) • 1-27 table
- Exiting
 - DEC/Test Manager subsystem • CD-11
 - Review subsystem • CD-125
- Expression • CD-6
 - collection-expression • CD-7
 - group-expression • CD-7
 - object-expression • CD-7
 - result-description-expression • CD-7
 - syntax of • CD-6
 - test-expression • CD-7

Expression (cont'd.)

- test-group-expression • CD-8
- variable-expression • CD-8
- EXTRACT • 1-23 table, CD-73
 - file-spec • CD-74
 - format of • B-16
 - /LOG • CD-73
 - /NOLOG • CD-73
 - /TERMINATION_CHARACTER • CD-73, B-16
- Extracting an INPUT file
 - See INPUT file, extracting from a SESSION file

F

- File-name parameter • CD-3
- File protection • 4-14
 - @file-spec • 6-32
- File-spec parameter • CD-2
- File-type • CD-3
- Filter • 6-35, 6-35 table
- Filter, Analyzer
 - See Performance and Coverage Analyzer, Analyzer filter
- Filters • 3-2 table
 - adding to a new test description • 6-35 table
 - adding to an existing test description • 6-35 table
 - adding to a test description • 6-35 table, CD-44, CD-87, CD-88
- Analyzer
 - See Performance and Coverage Analyzer, Analyzer filter
 - caution about using • 6-36
 - displaying the filters associated with a test description • 6-36, CD-184
 - removing from a test description • 6-36, CD-88
 - replacing for a test description • CD-87
 - using DTM\$RESULT • 6-11
 - when executed • 6-2, 6-35
- FIRST (Review subsystem command) • 5-7-List, CD-126
- FMS
 - interactive testing example • C-1 to C-9
- Forms Management System
 - See FMS

G

/GROUP • CD-10

Group-expression • CD-6 list, CD-7

Group-name • CD-4

Groups • 3-2 table, 6-13

changing the hierarchy of • CD-104

creating • 6-15, CD-40, CD-78

restrictions on • CD-78

creating a group hierarchy • 6-14, 6-17, CD-78

example • 6-22

creating from the Review subsystem

canceling creation of • 5-21

inserting test descriptions • 5-22

names for • 5-22

deleting • 6-22, CD-62

description of • 6-13

displaying • 6-19

contents of a group • CD-62

information about • CD-174

memberships of a group • CD-62

inserting

differences between inserting groups and tests into a group • CD-81

groups into a group • 6-14, 6-17, CD-40, CD-78

restriction on • 6-14

test descriptions into a group • 6-15,

CD-40, CD-81

example • 6-16

when to use the INSERT GROUP command • 6-18

when to use the INSERT TEST_ DESCRIPTION command • 6-17

modifying • CD-84

organizing tests by common attributes • 6-13

removing

groups from a group • 6-20, CD-62, CD-104

remark from • CD-83

subgroups from a group • 6-20, CD-104

test description from a group • 6-20

test descriptions from a group • 6-15, CD-62, CD-107

replacing the remark • CD-83, CD-84

restrictions when creating • CD-78

subgroups • 6-14

Groups (cont'd.)

using to specify groups of tests • 6-13

verifying the contents of a group • 6-18, 6-19

H

HELP • 1-21 table, CD-76

DEC/Test Manager subsystem • CD-76

displaying • 1-4, CD-76

online • 1-4

Review subsystem • CD-127

topic • CD-76

HELP (Review subsystem command)

topic • CD-127

History file • 6-36

adding a remark to • CD-101

commands logged in • CD-177-List

commenting unusual occurrence in • CD-101

deleting information from • CD-64

displaying • 6-36, CD-177

logging commands in • 6-36, 6-37

logging unusual events in • 6-37

remarks associated with commands • 6-37

I

Initialization command file • 6-34, CD-57

See also Command file

example • 6-34

executing • 6-34

keypad definitions in • CD-123

INPUT file

contents of • B-1, B-11

creating • B-11

creating by editing an existing test script • B-16

creating with a text editor • B-16, B-17

example • B-17

description of • B-1, B-11

entering comments in • B-15

example of • B-11

extracting from a SESSION file • CD-74, B-13, B-16

description of process • B-17

example • B-17

translations performed • B-13

in CMS library

restriction on • B-16

INPUT file (cont'd.)

- problems recording a SESSION file on a VT100 series terminal • B-21
- problems with typeahead when recording a SESSION file from an INPUT file • B-21
- recording a SESSION file • B-18
 - using CREATE TEST_DESCRIPTION/INPUT • B-18
 - using MODIFY TEST_DESCRIPTION/INPUT • B-18
 - using the INSERT recording function • B-19
- recording a SESSION file from • CD-74
- restriction on nesting • B-20
- special strings • B-12
 - See also Special strings
 - special strings in • CD-73
 - terminal characteristics when extracting an INPUT file from a SESSION file • B-20
 - terminal characteristics when recording a SESSION file from an INPUT file • B-20
 - translations for recording functions when extracting • CD-73
 - translations when extracting • CD-74
 - using a nondefault termination character • B-16
 - using to edit a SESSION file • B-16
- INSERT (Review subsystem command) • 5-22, CD-128
 - /COMPARISON_ABORTED • CD-128
 - comparison status qualifiers • 5-22
 - /CONFIRM • CD-128
 - /NEW • CD-128
 - /NOCONFIRM • CD-128
 - /NOT_RUN • CD-128
 - result-description-expression • CD-129
 - /SUCCESSFUL • CD-129
 - /UNSUCCESSFUL • CD-129
 - /UPDATED • CD-129
- INSERT GROUP • 1-24 table, 6-17, CD-40, CD-77
 - /CONFIRM • CD-77
 - group-expression • CD-77
 - /LOG • CD-77
 - /NOCONFIRM • CD-77
 - /NOLOG • CD-77
 - remark • CD-78
 - when to use • 6-18
- INSERT TEST_DESCRIPTION • 1-24 table, 6-15, CD-40, CD-80

INSERT TEST_DESCRIPTION (cont'd.)

- /CONFIRM • CD-80
- /GROUP • CD-80
- group-expression • CD-81
- /LOG • CD-80
- /NOCONFIRM • CD-80
- /NOLOG • CD-80
- remark • CD-81
- /TEST_DESCRIPTION • CD-81
- test-group-expression • CD-81
- when to use • 6-17
- Interactive and batch environments
 - effects on collection execution • 4-9
- Interactive terminal session
 - See also Test, interactive editing
 - See SESSION file, editing
 - example • 3-14
 - recording • 1-11, CD-89, B-2 example, C-1
 - terminating • 1-14, 3-11
 - example • C-5
- Interactive test
 - See Test, interactive
- Invoking DEC/Test Manager
 - from a command file • CD-15
 - from batch • CD-15
- Invoking DEC/Test Manager as a subsystem • 1-5
- Invoking the DEC/Test Manager subsystem • CD-15
- Invoking the Review subsystem • CD-109

K

Keypads

- DEC/Test Manager
 - defining keys • CD-57
 - display defaults • CD-57
- DEFINE/KEY • 6-27
- DISPLAY/BENCHMARK • 3-18
 - key definitions • 3-18 table
- Review subsystem • 5-25
 - changing key definitions • 5-27
 - default key definitions • 5-25 figure
 - defining keys • CD-123
 - displaying defaults • CD-123
 - saving key definitions • 6-28

Keypads (cont'd.)

SHOW/BENCHMARK

key definitions • 5-11 table

SHOW/DIFFERENCE

key definitions • 5-14 table

SHOW/RESULT

key definitions • 5-11 table

using • 6-26

L

LAST (Review subsystem command) • 5-7-List, CD-131

Leaving

DEC/Test Manager subsystem • CD-11

Library

access

establishing • 2-5

establishing group access with ACLs • 2-5

establishing group access with a UIC

mask • 2-5

required • 2-4

suggested • 2-4

checking for errors in • 6-31, CD-202

commands

for creating • 1-21 table

for using • 1-22 table

converting to current version • CD-25

converting to the current version • 2-9

correcting errors in • 6-31, CD-201, CD-202

creating • 1-7, 1-21 table, 2-1 to 2-10, CD-42

example • C-1

restrictions on • CD-42

current • CD-161

description of • 1-6, 2-1

displaying

current • 2-10

directory-spec for • CD-180

summary information about • 6-38,

CD-166

errors in • CD-202

extraneous subdirectories • CD-202

files, access required for • 2-4

logical name for

See DTM\$LIB

recovering • 6-32, CD-202

selecting • 1-7, 2-10, CD-161

Library

selecting (cont'd.)

without verification • 2-10, CD-160

specifying a UIC protection mask • 2-5

structure of • 2-1 figure, 2-3

using • 1-22

verifying • 6-31, CD-202

/LOG • CD-9

Log file • 3-5

M

Manual screen compare

See Compare, manual

Messages • A-1

display of • A-1

fields in • A-1

format of • A-1

list of • A-2 to A-50

name • A-1

severity code • A-1, A-2

error • A-2

fatal • A-2

informational • A-2

success • A-2

warning • A-2

text • A-1

MODIFY GROUP • 1-24 table, CD-83

group-expression • CD-84

/LOG • CD-83

/NOLOG • CD-83

/NOREMARK • CD-83

/REMARK • CD-83

remark • CD-84

remark parameter and remark qualifier • CD-4

Modifying groups

See Groups, modifying

Modifying test descriptions

See Test description, modifying

Modifying variables

See Variables, modifying

MODIFY TEST_DESCRIPTION • 1-23 table, 3-20,

CD-69, CD-85

/AUTO_COMPARE • CD-85

/BENCHMARK • CD-86

/EPILOGUE • 3-29, CD-87

/FILTER • 6-35 table, CD-88

MODIFY TEST_DESCRIPTION (cont'd.)

/GROUP • CD-91
/INPUT • CD-74, CD-88
/INTERACTIVE • CD-88
/LOG • CD-89
/NOAUTO_COMPARE • CD-85
/NOBENCHMARK • CD-86
/NOEPILOGUE • 3-29, CD-87
/NOFILTER • 6-36, CD-88
/NOINTERACTIVE • CD-88
/NOLOG • CD-89
/NOPROLOGUE • 3-29, CD-89
/NOREMARK • CD-90
/NOVARIABLE • CD-91
/PROLOGUE • 3-29, CD-89
/RECORD • CD-89
/REMARK • CD-90
remark • CD-92
remark parameter and remark qualifier • CD-4
/TEMPLATE • CD-90
/TERMINATION_CHARACTER • CD-90
/TEST_DESCRIPTION • CD-91
test-group-expression • CD-92
/VARIABLE • CD-91

MODIFY VARIABLE • 1-24 table, 6-5, CD-94
/GLOBAL • CD-94
/LOCAL • CD-94
/LOG • CD-94
/LOGICAL • CD-94
/NOLOG • CD-94
/NOREMARK • CD-95
/NUMERIC • CD-95
/REMARK • CD-95
remark • CD-96
remark parameter and remark qualifier • CD-4
/STRING • CD-95
/SYMBOL • CD-95
/VALUE • CD-95
variable-expression • CD-95

N

NEXT (Review subsystem command) • 5-8,
CD-132
/COMPARISON_ABORTED • CD-132
comparison status qualifiers • 5-8
count • CD-133

NEXT (Review subsystem command) (cont'd.)

/NEW • CD-132
/NOT_RUN • CD-132
/SUCCESSFUL • CD-132
/UNSUCCESSFUL • CD-133
/UPDATED • CD-133

O

Object-expression • CD-6 list, CD-7
Online help
See HELP
/OUTPUT • CD-9
Output file qualifiers • 5-9, CD-111, CD-138,
CD-144
/BENCHMARK • CD-138, CD-144
description of • CD-138, CD-144
/DIFFERENCE • CD-138, CD-144
on the PRINT command • CD-138
on the SHOW command • CD-144
/RESULT • CD-138, CD-144
Output files • 5-3 list
benchmark file • 5-3 list
description of • 5-3
difference file • 5-3 list
printing • 5-16
result file • 5-3 list
selecting with output file qualifiers • 5-5

P

Parameter • CD-1, CD-1 list
collection-expression • CD-6 list, CD-7, CD-59,
CD-170
collection-name • CD-2, CD-22, CD-36, CD-99,
CD-108, CD-154, CD-196, CD-199
restrictions on • CD-2
command • CD-148, CD-194
command-string • CD-57, CD-123
count • CD-120, CD-133
directory-spec • CD-3, CD-25, CD-41, CD-156,
CD-161, CD-164
logical names for • CD-3
restrictions on • CD-3
expression • CD-6
• CD-6 list
restrictions on • CD-6

Parameter

- expression (cont'd.)
 - syntax of • CD-6
 - file-name • CD-3
 - restrictions on • CD-3
 - file-spec • CD-2, CD-17, CD-74, CD-97, CD-116, CD-158, CD-162, B-16
 - restrictions on • CD-2
 - group-expression • CD-6 list, CD-7, CD-61, CD-77, CD-78, CD-81, CD-84, CD-103, CD-106, CD-173
 - group-name • CD-4, CD-39
 - key-name • CD-56, CD-122
 - name requirements • CD-3
 - object-expression • CD-6 list, CD-7, CD-177
 - process-name • CD-18, CD-117
 - remark • CD-1, CD-4, CD-30, CD-36, CD-39, CD-41, CD-48, CD-53, CD-60, CD-61, CD-64, CD-65, CD-68, CD-78, CD-81, CD-84, CD-92, CD-96, CD-99, CD-101, CD-104, CD-106, CD-108, CD-151, CD-154, CD-156, CD-158, CD-162, CD-164, CD-196, CD-199, CD-201
 - using quotation marks in • CD-4
 - result-description-expression • CD-6 list, CD-7, CD-129, CD-137, CD-144, CD-150
 - restrictions on • CD-6, CD-7
 - result-description-name • 5-2, CD-4, CD-139
 - test-expression • CD-6 list, CD-7, CD-65
 - test-group-expression • 4-3, CD-1, CD-6 list, CD-8, CD-36, CD-81, CD-92, CD-106, CD-186
 - using parameter qualifiers • CD-8
 - test-name • 3-2, 3-17, CD-5, CD-29, CD-30, CD-48, CD-70
 - topic • CD-76, CD-127
 - variable-expression • CD-6 list, CD-8, CD-68, CD-95, CD-189
 - variable-name • CD-5, CD-52
 - restriction on • 6-4
 - restrictions on • CD-5
 - variable-value • 6-4, CD-5, CD-53
 - restrictions on • CD-5
 - using an equal sign with • CD-5
 - using quotation marks with • CD-5
- Partially run collections
- See Collections, partially run
- PCA (Review subsystem command) • CD-134
- See also Performance and Coverage Analyzer
- Percent sign (%) • CD-8
- Performance and Coverage Analyzer • 5-2
- See also PCA (Review subsystem command) Analyzer • 7-12
- invoking from the Review subsystem • 7-8
 - Analyzer filter • 7-9, CD-134
 - analyzing data
 - by test • 7-9
 - for all tests • 7-9
 - Collector • 7-3
 - Collector commands • 7-3
 - Collector initialization file • 7-3, 7-4, 7-10
 - description of • 7-1
 - example • 7-10
 - invoking • CD-134
 - system considerations • 7-12
 - using with DEC/Test Manager • 7-1
- PLAY • 1-23 table, 3-19, CD-97
- /DISPLAY_TERMINAL • CD-97
 - file-spec • CD-97
 - /LOG • CD-97
 - /NOLOG • CD-97
 - /NORESULT_FILE • CD-97
 - /RESULT_FILE • CD-97
 - terminal characteristics with • 3-19
 - terminal type with • 3-19
- PRINT (Review subsystem command) • 5-9, 5-16, CD-135
- /BENCHMARK • CD-135
 - /COMPARISON_ABORTED • CD-135
 - comparison status qualifiers • 5-16
 - /DIFFERENCE • CD-135
 - /LOG • CD-136
 - /NEW • CD-136
 - /NOLOG • CD-136
 - /NOT_RUN • CD-136
 - /NOW • CD-136
 - output file qualifiers • 5-3 list, 5-17
 - possible errors for read-only reviewers • 5-6
 - /RESULT • CD-136
 - result-description-expression • CD-137, CD-150
 - /SELECTED • CD-136
 - /SUCCESSFUL • CD-136
 - /UNSUCCESSFUL • CD-137
 - /UPDATED • CD-137

Process

- attaching to • CD-18, CD-118
- spawning • CD-195

Prologue file

- collection • 1-15, 6-3
 - canceling the default • 3-29, 6-3, CD-163
 - description of • 6-2
 - displaying the default • CD-181
 - establishing the default • 3-29, 6-3, CD-163
 - (for use with the Performance and Coverage Analyzer) • 7-4, 7-10
 - overriding the default • 3-29, 6-3, CD-163
 - storing • 3-26
 - storing in a CMS library • 3-30
 - when executed • 6-2
- description of • 6-1
- storing • 2-3, 3-30
- test • 1-8, 3-2 table, 3-4, 6-2
 - adding to a test description • 3-29, 6-2, CD-89
 - canceling • 3-29
 - canceling for a test description • 3-20, CD-89
 - description of • 3-4, 6-2
 - removing from a test description • 3-20, 6-3, CD-89
 - replacing for a test description • 3-20, 3-29, CD-89
 - storing • 3-26
 - storing in a CMS library • 3-30
 - uses of • 6-2
 - when executed • 6-2
- using • 6-1

Q

Qualifier

- command • CD-1, CD-9
 - negated • 3-20, 3-22
 - placement • CD-1
 - syntax • CD-9
- comparison status
 - See Comparison status qualifiers
- output file
 - See Output file qualifiers
- parameter • CD-1, CD-9, CD-10, CD-36

Qualifier

- parameter (cont'd.)
 - default • CD-10
 - /GROUP • CD-10
 - placement • CD-1
 - /TEST_DESCRIPTION • CD-10
- using the DCL SUBMIT command qualifiers • CD-199 list

R

Recording an interactive terminal session • 3-8

- See also Test, interactive example • 3-14, C-1
- restrictions on • 3-12
 - REGIS graphics • 3-14
 - timing-dependent applications • 3-12
 - unsolicited input • 3-13
 - using CTRL/Y • 3-13
 - using type-ahead • 3-13
 - VAXstations with UIS • 3-14
- specifying manual screen compare • 3-11

Recording an interactive test

- See Recording an interactive terminal session
- See Test, interactive

Recording functions • 3-10 table, CD-73

- ABORT (CTRL/P CTRL/C) • 3-11
- BEGIN_COMMENT (CTRL/P !) • 3-10 table, 3-12
- BEGIN_COMPARE (CTRL/P B) • 3-10 table, 3-11
- COMPARE_SCREEN (CTRL/P C) • 3-10 table, 3-11
- DELAY (CTRL/P D) • 3-10 table, 3-12
 - restriction on • 3-12
- description of • 3-9
- END_COMPARE (CTRL/P E) • 3-10 table, 3-11
- HELP (CTRL/P ?) • 3-10 table, 3-11
- INSERT
 - using • B-19
- INSERT (CTRL/P I) • 3-10 table, 3-12
- invalid • 3-10
- QUIT (CTRL/P CTRL/Z) • 3-11
- TERMINATE (CTRL/P CTRL/P) • 3-11
- termination character • 3-10
- translation of • B-16
- using • 3-9

Recording functions (cont'd.)

WAIT (CTRL/P W) • 3-10 table, 3-12
restriction on • 3-12

Recovering a library with errors

See Library, recovering

RECREATE • 1-26 table, 4-14, CD-37, CD-99,
CD-199 list

collection_name • CD-99

/CONFIRM • CD-99

/LOG • CD-99

/NOCONFIRM • CD-99

/NOLOG • CD-99

remark • CD-99

Recreating collections

See Collections, recreating

Regression testing

definition of • 1-1

purpose of • 1-1

with DEC/Test Manager • 1-2 figure, 1-3

without DEC/Test Manager • 1-2

REMARK • CD-4, CD-101

/LOG • CD-101

/NOLOG • CD-101

remark • CD-101

Remark • 1-7, 1-10, 3-2 table, CD-1, CD-4

command • CD-4

on a group

removing • CD-83

replacing • CD-83

on a test description

removing • CD-90

replacing • CD-90

parameter • CD-92, CD-96

qualifier • CD-90, CD-95

remark parameter and remark qualifier • CD-84,
CD-92, CD-96

using quotation marks in • CD-4

Remark parameter • CD-4, CD-30, CD-84

Remark qualifier • CD-4, CD-29, CD-83

REMOVE GROUP • 1-24 table, 6-20, 6-21, CD-62,
CD-103

/CONFIRM • CD-103

group-expression • CD-103

/LOG • CD-103

/NOCONFIRM • CD-103

/NOLOG • CD-103

remark • CD-104

REMOVE TEST_DESCRIPTION • 1-24 table, 6-20,
CD-62, CD-66, CD-105

/CONFIRM • CD-105

/GROUP • CD-106

group-expression • CD-106

/LOG • CD-105

/NOCONFIRM • CD-105

/NOLOG • CD-105

remark • CD-106

/TEST_DESCRIPTION • CD-106

test-group-expression • CD-106

Removing groups from a group

See Groups, removing

Removing subgroups from a group

See Groups, removing

Removing test descriptions from a group

See Test description, removing

RES_SCREEN file

See Result file, printable screens file

Result description • 1-18, 4-2, CD-109

contents of • 5-2

current • 5-7-List

description of • 5-2

displaying • 5-9

example • 5-2

format of • 5-2

identifying • 5-5

by comparison status • 5-5

by result-description-name • 5-5

identifying by comparison status • 5-5

selecting • 5-7-List

by common characteristics • 5-5

with the positioning commands • 5-7-List

Result-description-expression • CD-6 list, CD-7,
CD-110, CD-111

Result-description-name • 1-18, 5-2, 5-8, CD-4,
CD-109, CD-110

Result file • 1-15, 3-5, 4-2, 4-9, 5-1, CD-23

comparing with the benchmark file • 4-2, 4-10

comparison to a batch log file • 4-2

deleting • CD-66

description of • 4-2, 5-3 list

displaying • CD-142

displaying for an interactive test • 5-11

examining from the Review subsystem • 5-1

printable screens file (test-name.RES_

SCREENS) • 5-3 list

Result file (cont'd.)

RES_SCREEN file

See Result file, printable screens file

restriction on printing • 5-17

REVIEW • 5-6, CD-108

collection-name • CD-108

example • C-7

prompt • CD-108

/READ_ONLY • CD-108

remark • CD-108

Reviewer

primary • CD-108

read-only • CD-108

Reviewing test results • CD-109

Review subsystem • 1-18, 4-12, 7-12

benchmark file

displaying • CD-140

updating • CD-151

canceling commands • 5-25

collection summary • CD-110

redisplaying • CD-110

commands

ATTACH • CD-117 to CD-118

BACK • CD-119 to CD-120

by function • 5-7 list, CD-109 list

DEFINE/KEY • CD-121 to CD-124

EXIT • CD-125

@file-spec • CD-116

FIRST • CD-126

for manipulating result descriptions • 1-28

table

for selecting result descriptions • 1-28

table

general • 1-27 table

HELP • CD-127

INSERT • CD-128 to CD-130

LAST • CD-131

NEXT • CD-132 to CD-133

PCA • CD-134

PRINT • CD-135 to CD-138

SELECT • CD-139

SHOW • CD-140 to CD-145

SPAWN • CD-146 to CD-149

UPDATE • CD-150 to CD-152

with qualifiers • CD-112 list

comparison status • CD-110

comparison status qualifiers • CD-110, CD-111,

CD-138, CD-144

Review subsystem (cont'd.)

creating a group on exit • 5-21, CD-129

default name for group • CD-130

restriction on • 5-22

creating a new benchmark file • CD-151

difference file

displaying • CD-141

displaying help • CD-127

displaying information about the collection •

CD-144

displaying the collection summary information •

CD-143

displaying the output files for a collection •

CD-144

examining performance and coverage data •

CD-112

exiting • 5-21, 5-24, CD-125

/FILES qualifier • CD-145

grouping tests • CD-129

invoking • 5-6, CD-109

as a read-only reviewer • CD-109

as the primary reviewer • CD-109

example • C-7

in read-only mode • CD-108

invoking the Performance and Coverage

Analyzer from • CD-112, CD-134

See also Performance and Coverage

Analyzer

keypad

defining keys • CD-123

displaying defaults • CD-123

help • CD-123

replacing defaults • CD-123

moving

backward • CD-120

forward • CD-133

to a specified result description • CD-139

to the first result description • CD-126

to the last result description • CD-131

output file qualifiers • CD-138, CD-144

overview of • 5-1

positioning commands

BACK • 5-8

FIRST • 5-7 list

LAST • 5-7 list

NEXT • 5-8

SELECT • 5-8

Review subsystem (cont'd.)

- primary reviewer • 5-6
 - printing files
 - all currently selected files • CD-137
 - all previously selected files • CD-137
 - immediately • CD-137
 - on exit from the Review subsystem • CD-137
 - read-only mode • 5-6, CD-108
 - disappearance of files selected for printing • 5-6
 - possible errors in the Collection Summary Information • 5-6
 - possible errors in the SHOW command output • 5-6
 - read-only reviewer • 5-6
 - referencing a result description • CD-110 to CD-111
 - result description
 - general information about describing • 5-5
 - referencing • CD-110 to CD-111
 - selecting • 5-7-List
 - result file
 - displaying • CD-142
 - reviewer
 - primary • 5-6
 - read-only • 5-6
 - reviewing a collection • CD-109
 - reviewing a partially run collection • 5-23
 - reviewing performance and coverage data • CD-134
 - See also, PCA (Review subsystem command)
 - See also, Performance and Coverage Analyzer
 - selecting a result description • 5-7-List, CD-110
 - spawning a subprocess • CD-149
 - /SUMMARY qualifier • CD-145
 - terminating • CD-125
 - terminology • 5-2
 - typing a benchmark file • CD-140
 - typing a difference file • CD-141
 - typing a result file • CD-142
 - updating a benchmark file • CD-151
- RUN** • 1-26 table, 4-9, CD-37, CD-153
- collection-name • CD-154
 - /CONFIRM • CD-153

RUN (cont'd.)

- example • C-6
- executed from batch • CD-154, CD-155
- /LOG • CD-153
- /LOG_FILE • 4-9, CD-153
- /NOCONFIRM • CD-153
- /NOLOG • CD-153
- /NOLOG_FILE • CD-154
- /NOOUTPUT • CD-154
- /OUTPUT • CD-154
- remark • CD-154
- with large collections • 4-9, CD-154
- with test-name • 4-9, CD-155

S

- Screen compare
 - See Compare
- Screens
 - printable files
 - test-name.BMK_SCREEN • 5-3 list
- SELECT (Review subsystem command) • 5-8, CD-139
 - example • C-7
 - result-description-name • CD-139
- SESSION file • 1-12, 1-13, 3-3, CD-74, CD-89
 - braces in • CD-74
 - contents of • B-1
 - creating
 - example • C-1, C-6
 - CTRL/A in • B-10
 - description of • B-1
 - editing using an INPUT file • B-16
 - effects of asynchronous events on • B-6
 - effects of program output on • B-6
 - effects of system response on • B-4, B-5
 - effects of terminal driver on • B-4, B-6
 - effects of timing on • B-4
 - effects of typing speed on • B-5
 - example • B-2
 - executing • 3-19
 - extracting an INPUT file • B-16
 - description of process • B-17
 - example • B-17
 - extracting an INPUT file from • B-16
 - first record • B-6
 - in CMS library • B-16

SESSION file (cont'd.)

- mixed input and output • B-4
- modifying • B-16
- modifying directly • B-9
 - BEGIN_COMPARE records • B-10
 - COMPARE_SCREEN records • B-10
 - END_COMPARE records • B-10
 - INPUT records • B-10
 - OUTPUT records • B-10
 - TIMING records • B-11
 - WAIT records • B-11
- problems recording a SESSION file on a VT100 series terminal • B-21
- problems with typeahead when recording a SESSION file from an INPUT file • B-21
- recording
 - example • B-2
- recording from an INPUT file • CD-74, B-16, B-18
 - translations performed • B-13
 - using CREATE TEST_DESCRIPTION/INPUT • B-18
 - using MODIFY TEST_DESCRIPTION/INPUT • B-18
 - using the INSERT recording function • B-19
- records
 - description of • B-7
 - structure of • B-2, B-6
- record type • B-7, B-7 table
 - BEGIN_COMMENT (I) • B-8
 - BEGIN_COMPARE (B) • B-8, B-10
 - COMPARE_SCREEN (C) • B-8, B-10
 - DELAY (D) • B-8, B-9
 - END_COMPARE (E) • B-8, B-10
 - INPUT (I and 1) • B-8, B-10
 - OUTPUT (O and 0) • B-8, B-10
 - TIMING (T) • B-8, B-9, B-11
 - WAIT (W) • B-8, B-9, B-11
- restrictions on modifying directly • B-9
- special strings • B-1
- structure • B-4
- terminal characteristics block • B-2, B-6 figure
 - caution about • B-7
 - contents of • B-7
 - description of • B-6
- terminal characteristics in • CD-74
- terminal characteristics when extracting an INPUT file from a SESSION file • B-20

SESSION file (cont'd.)

- terminal characteristics when recording a SESSION file from an INPUT file • B-20
- timing information
 - restrictions on changing • B-11
 - verifying • 3-19
- SET BENCHMARK_DIRECTORY • 1-25 table, CD-156
 - directory-spec • CD-156
 - /LOG • CD-156
 - /NOLOG • CD-156
 - remark • CD-156
- SET EPILOGUE • 1-25 table, 3-29, CD-158
 - file-spec • CD-158
 - /LOG • CD-158
 - /NOLOG • CD-158
 - remark • CD-158
- SET LIBRARY • 1-21 table, 2-10, CD-160
 - directory-spec • CD-161
 - /LOG • CD-160
 - /NOLOG • CD-160
 - /NOVERIFY • 2-10, CD-160
 - selecting a library quickly • 2-10
 - /VERIFY • CD-160
- SET LIBRARY/NOVERIFY • 2-10
- SET NOBENCHMARK_DIRECTORY • 3-27, CD-157
 - remark • CD-156
- SET NOEPILOGUE • CD-159
 - remark • CD-158
- SET NOPROLOGUE • CD-163
 - remark • CD-162
- SET NOTEMLATE_DIRECTORY • 3-27, CD-165
 - remark • CD-164
- SET PROLOGUE • 1-25 table, 3-29, CD-162
 - file-spec • CD-162
 - /LOG • CD-162
 - /NOLOG • CD-162
 - remark • CD-162
- SET TEMPLATE_DIRECTORY • 1-25 table, CD-164
 - directory-spec • CD-164
 - /LOG • CD-164
 - /NOLOG • CD-164
 - remark • CD-164
- SHOW (Review subsystem command) • 5-9, CD-140
 - /BENCHMARK • 5-11, CD-140
 - /COMPARISON_ABORTED • CD-141

SHOW (Review subsystem command) (cont'd.)

comparison status qualifiers • 5-9 list
/DIFFERENCE • 5-13, CD-141
/DIFFERENCES
 example • C-7
/FILES • 5-10
/NEW • CD-141
/NOT_RUN • CD-142
/NOW • 5-17
/OUTPUT • CD-142
output file qualifiers • 5-3 list, 5-10
possible errors for read-only reviewers • 5-6
/RESULT • 5-11, CD-142
result-description-expression • CD-144
/SELECTED • 5-17
/SUCCESSFUL • CD-143
/SUMMARY • 5-10, CD-143
/UNSUCCESSFUL • CD-143
/UPDATED • CD-143
SHOW ALL • 1-25 table, CD-166
 /OUTPUT • CD-166
SHOW BENCHMARK_DIRECTORY • 1-25 table,
 CD-167
 /OUTPUT • CD-167
SHOW COLLECTION • 1-26 table, 4-12, CD-168
 /BENCHMARK_DIRECTORY • CD-168
 /BRIEF • CD-168
 /CLASS • CD-168
 collection-expression • CD-170
 example • C-6
 /FULL • CD-169
 /INTERMEDIATE • CD-169
 /OUTPUT • CD-169
 /TEMPLATE_DIRECTORY • CD-169
SHOW EPILOGUE • 1-25 table, CD-171
 /OUTPUT • CD-171
SHOW GROUP • 1-24 table, 6-19, CD-172
 /BRIEF • CD-172
 /CONTENTS • CD-62, CD-172
 /FULL • CD-173
 group-expression • CD-173
 /INTERMEDIATE • CD-173
 /MEMBER • CD-62, CD-173
 /NOCONTENTS • CD-173
 /OUTPUT • CD-173
SHOW HISTORY • CD-175
 /BEFORE • CD-175
 /NOTRANSACTION • CD-176

SHOW HISTORY (cont'd.)

object-expression • CD-177
/OUTPUT • CD-175
/SINCE • CD-175
/TRANSACTION • CD-176
/USER • CD-176
SHOW LIBRARY • 1-21 table, 2-10, CD-180
 /OUTPUT • CD-180
SHOW PROLOGUE • 1-25 table, CD-181
 /OUTPUT • CD-181
SHOW TEMPLATE_DIRECTORY • 1-25 table,
 CD-182
 /OUTPUT • CD-182
SHOW TEST_DESCRIPTION • 1-23 table, 3-25,
 CD-183
 /BENCHMARK • CD-183
 /BRIEF • CD-184
 /COMPARISON_TYPE • CD-184
 /EPILOGUE • CD-184
 example • C-5
 /FILTER • 6-36, CD-184
 /FULL • 6-36, CD-184
 /GROUP • CD-186
 /GROUPS • CD-66, CD-184
 /INTERMEDIATE • CD-185
 /OUTPUT • CD-185
 /PROLOGUE • CD-185
 /REMARK • CD-185
 /TEMPLATE • CD-185
 /TEST_DESCRIPTION • CD-186
 test-group-expression • CD-186
 /TYPE • CD-185
 /VARIABLE • CD-185
SHOW VARIABLE • 1-24 table, CD-188
 /BRIEF • CD-188
 /FULL • CD-188
 /INTERMEDIATE • CD-188
 /OUTPUT • CD-188
 /REMARK • CD-189
 /SCOPE • CD-189
 /TESTS • CD-189
 /TYPE • CD-189
 /USAGE • CD-189
 /VALUE • CD-189
 variable-expression • CD-189
SHOW VERSION • 1-21 table, CD-191
 /OUTPUT • CD-191

SPAWN • 6-34, CD-192

/CARRIAGE_CONTROL • CD-192
/CLI • CD-192
command • CD-194
/INPUT • CD-193
/KEYPAD • CD-193
/LOGICAL_NAMES • CD-193
/NOCARRIAGE_CONTROL • CD-192
/NOCLI • CD-192
/NOKEYPAD • CD-193
/NOLOGICAL_NAMES • CD-193
/NONOTIFY • CD-193
/NOSYMBOLS • CD-194
/NOTIFY • CD-193
/NOWAIT • CD-194
/OUTPUT • CD-193
/PROCESS • CD-193
/PROMPT • CD-194
/SYMBOLS • CD-194
/WAIT • CD-194

SPAWN (Review subsystem command) • CD-146

/CARRIAGE_CONTROL • CD-146
/CLI • CD-146
command • CD-148
/INPUT • CD-147
/KEYPAD • CD-147
/LOGICAL_NAMES • CD-147
/NOCARRIAGE_CONTROL • CD-146
/NOCLI • CD-146
/NOKEYPAD • CD-147
/NOLOGICAL_NAMES • CD-147
/NONOTIFY • CD-147
/NOSYMBOLS • CD-148
/NOTIFY • CD-147
/NOWAIT • CD-148
/OUTPUT • CD-147
/PROCESS • CD-147
/PROMPT • CD-148
/SYMBOLS • CD-148
/WAIT • CD-148

Special strings • CD-73, CD-74

DEC/Test Manager processing during
translation • B-15
description of • B-12
for 8-bit control characters
formats of • B-13
translations for • B-30 table

Special strings (cont'd.)

for arrow key codes
formats of • B-14
translations for • B-36 table
for control characters
formats of • B-13
translations for • B-27 table
for editing key codes
formats of • B-14
translations for • B-33 table
for function key codes
formats of • B-14
translations for • B-31 table
for key names • B-14
arrow keys • B-14
editing keys • B-14
function keys • B-14
keypad keys • B-14
for keypad key codes
formats of • B-14
translations for • B-34 table
format of • B-12
for nonprinting characters
formats of • B-13
translations for • B-27 table
for recording functions
formats of • B-14
translations for • B-37 table
in SESSION file • B-1
rules for using in an INPUT file • B-15
case of text • B-15
entering comments • B-15
entering return characters • B-15
including braces in text • B-15
INSERT recording function • B-15
record boundaries • B-15
using special string equivalents for
recording functions • B-15
translation of recording functions • B-16
translations
for 8-bit control characters
• B-30 table
for arrow key codes
• B-36 table
for control characters
• B-27 table
for editing key codes
• B-33 table

Special strings

translations (cont'd.)

for function key codes

- B-31 table

for keypad key codes

- B-34 table

for nonprinting characters

- B-27 table

for recording functions

- B-37 table

types recognized by DEC/Test Manager •

B-12 list

untranslatable • B-20

using a nondefault termination character • B-16

State-name

DTM • CD-57

GOLD_DTM • CD-57

GOLD_REVIEW • CD-123

REVIEW • CD-123

STOP • 1-26 table, 4-15, 5-23, 5-24, CD-23,
CD-196

collection-name • CD-196

/CONFIRM • CD-196

/LOG • CD-196

/NOCONFIRM • CD-196

/NOLOG • CD-196

remark • CD-196

Stopping collections

See Collections, stopping execution of

SUBMIT • 1-26 table, 4-4, 4-7, 4-8, CD-37,
CD-154, CD-198,
CD-199 list

collection-name • CD-199

/CONFIRM • CD-198

default qualifiers • 4-8

/KEEP • CD-198

/LOG • CD-198

/NOCONFIRM • CD-198

/NOKEEP • CD-198

/NOLOG • CD-198

remark • CD-199

using DCL SUBMIT command qualifiers •
CD-199 list

with large collections • CD-154

Subprocess

attaching to • 6-34

creating • 6-34

T

Template directory

canceling the default • 3-27, CD-165

displaying the default • 4-12, CD-182

establishing a hierarchy • 3-26

establishing the default • 3-26, CD-165

overriding the default • 3-26, CD-38, CD-165

Template file • 1-8, 3-2 table, 3-5, 3-8

See also Template directory

associating a file-spec with • CD-90

creating • 1-9, 3-3

default file-name • 3-2, CD-49

for an interactive test • 3-3

for a noninteractive test • 3-3

generalizing with variables • 3-3

naming • 3-3

replacing • 3-20, CD-90

replacing with the default • 3-20

storing • 2-3, 3-3, 3-26

in a CMS library • 3-27

using DTM\$TEST_NAME • 6-12

Terminal characteristics • 3-7, CD-74

when extracting an INPUT file from a SESSION
file • B-20

when recording an INPUT file from a SESSION
file • B-20

when recording a SESSION file from an INPUT
file • B-20

with PLAY • 3-19

Terminal characteristics block • B-2, B-4, B-6 figure

caution about • B-7

description of • B-6

Terminal type • 3-7

in terminal characteristics block • B-6

problems recording a SESSION file on a VT100
series terminal • B-21

when recording an INPUT file from a SESSION
file • B-20

with PLAY • 3-19

Terminating

DEC/Test Manager subsystem • CD-11

interactive terminal session • 3-11

Review subsystem • CD-125

Termination character • 1-14, 3-10, CD-73

default • CD-48, CD-73

in recording functions • 3-10

Termination character (cont'd.)

- replacing • CD-90
- specifying • 3-21
- using special string equivalents in INPUT files • B-15
 - when extracting an INPUT file • CD-73, B-16
 - with the EXTRACT command • B-16
- /TEST_DESCRIPTION • CD-10
- Test description • 1-8, 4-1
 - adding
 - epilogue file to • CD-87
 - filters to • 6-35 table, CD-88
 - prologue file to • CD-89
 - variable to • CD-91
 - canceling
 - epilogue file for • CD-87
 - prologue file for • CD-89
 - changing a variable value • CD-91
 - copying • 3-22, CD-30
 - with modified field values • 3-22
 - creating • 1-10, 3-6, CD-48
 - example • C-1
 - for an interactive test • 3-1
 - for a noninteractive test • 3-1
 - default benchmark file-name • 3-2, CD-44, CD-49
 - default template file-name • 3-2, CD-49
 - deleting • 3-25, CD-66
 - description of • 3-1
 - displaying • 3-24
 - contents of • CD-186
 - field values of • CD-186
 - filters associated with a test description • 6-36
 - example • 1-10
 - fields • 3-2, 3-2 table
 - benchmark • 3-2 table, 3-3
 - canceling values • 3-20
 - changing values • 3-20
 - epilogue • 3-4
 - filter • 3-2 table
 - group • 3-2 table
 - remark • 3-2 table
 - specifying values for • 3-6, 3-8
 - template • 3-2 table, 3-3
 - test epilogue • 3-2 table
 - test-name • 3-2, 3-2 table
 - test prologue • 3-2 table, 3-4

Test description

- fields (cont'd.)
 - variable • 3-2 table, 3-3
- field values
 - canceling • 3-20
 - changing • 3-20
- inserting into a group • 6-15, CD-81
 - example • 6-16
- marking
 - as interactive • CD-88
 - as noninteractive • CD-88
- missing files • 4-4, CD-49
- modifying • 3-20, CD-92
 - example • 3-21
 - specifying field values • CD-92
- modifying referenced files • 4-4
- related files • 3-1
- removing
 - all filters from • CD-88
 - benchmark file-spec from • CD-86
 - epilogue file from • CD-87
 - filters from • 3-21
 - from a group • 6-15, 6-20, CD-107
 - prologue file from • CD-89
 - remark from • 3-21, CD-90
 - variable from • CD-91
 - variables from • 3-21, CD-91
- replacing
 - benchmark file-spec • CD-86
 - epilogue file • CD-87
 - filters • 3-21
 - prologue file • CD-89
 - remark • 3-21, CD-90
 - template file-spec • CD-90
 - variables • 3-21
 - using logical names with • CD-50
- test description
 - overriding variable values • 6-7
- Test epilogue file
 - See Epilogue file, test
- Test-expression • CD-6 list, CD-7
- Test-group-expression • 3-20, 3-24, 4-3, 6-15, CD-6 list, CD-8
 - using parameter qualifiers • CD-10
- Test-name • 1-8, 3-2, 3-2 table, 3-6, 3-8, 3-17, CD-5

Test prologue file

See Prologue file, test

Test results

- accessing • 5-7 list
- displaying • 5-9
- evaluating • 5-1
- examining • 1-18, 4-6, 4-12, 5-9
- manipulating • 5-17
- printing output files • 5-16
- reviewing • 1-18, 1-26 table

Tests

- categorizing • 1-24
- commands
 - for categorizing • 1-24 table
 - for creating • 1-23 table
 - for executing • 1-26 table
 - for grouping • 1-24 table
 - for maintaining • 1-23 table
 - for organizing • 1-25 table
- comparing a benchmark file to a result file • 4-2
- comparing the results of test execution • 4-2, 4-10
- creating • 1-9, 1-23
- executing • 1-26, 4-1, 4-5, 4-6 figure
 - example • C-6
- executing with DEC/Test Manager • 4-1
- grouping • 1-24, 6-13, 6-14
- including in a collection • 4-3
- inserting into a group • 6-14
- interactive
 - comparing
 - character by character • CD-23
 - record by record • CD-24
 - screen by screen • CD-23
 - creating • 1-11, 3-7, 3-8
 - example • C-1
 - creating a benchmark file for • 4-2
 - creating a SESSION file for • 1-12
 - creating a test description for • 3-1, 3-8
 - default benchmark directory-spec • CD-89
 - default template directory-spec • CD-89
 - default template file-name • 3-2
 - displaying the benchmark file • 3-17, 5-11, CD-70
 - displaying the difference file • 5-10, 5-13
 - displaying the result file • 5-11
 - example • C-1 to C-9
 - FMS example • C-1

Tests

- interactive (cont'd.)
 - recording • 3-8, CD-89
 - recording the template file • 3-8
 - restrictions when creating • 3-12
 - running • CD-98
 - setting the screen size • 3-7
 - terminal characteristics • 3-7
 - terminal type • 3-7
 - terminating recording of • 3-11
 - test-name.BMK_SCREEN file • 5-3 list
 - test-name.RES_SCREEN file • 5-3 list
 - valid test description field values • 3-9
 - verifying the benchmark file • 3-17
 - maintaining • 1-23
 - marking as interactive • CD-88
 - marking as noninteractive • CD-88
 - noninteractive
 - creating • 1-8, 1-9, 3-4
 - creating a template file for • 1-9
 - creating a test description for • 1-10, 3-1, 3-6
 - default template file-name • 3-2
 - displaying output files • 5-10
 - example template file • 3-5
 - example test • 3-5
 - example test description • 3-6
 - valid test description field values • 3-6
 - writing the template file • 3-5
 - writing the test • 3-5
 - not run • 4-5
 - organizing • 1-8, 1-25
 - organizing by common attributes • 6-13
 - reviewing results • 1-18
 - running • 1-15
 - See Tests, executing
 - selecting • 4-3
- ## Test system
- creating
 - example • C-1
 - organizing • 3-1
 - example • C-1
- ## Timing interval
- recorded by DEC/Test Manager • B-9
- ## Translations performed
- translation tables • B-21 to B-37 table
 - when extracting an INPUT file from a SESSION file • B-13

Translations performed (cont'd.)

when recording a SESSION file from an INPUT file • B-13

U

UIC (user identification code) • 2-5
 protection mask • 2-5
 suggested mask for library • 2-5

UPDATE

warning about use • CD-152
UPDATE (Review subsystem command) • 5-18, CD-150
 /CONFIRM • CD-150
 /LOG • CD-150
 /NOCONFIRM • CD-150
 /NOLOG • CD-150
 remark • CD-151

User identification code
 See UIC

V

Variable-expression • CD-6 list, CD-8
Variable-name • CD-5
 restriction on • 6-4
Variables • 1-24 table, 3-2 table
 • 1-24 table
 adding to the DEC/Test Manager library • 6-4
 assigning a value to • 6-7
 changing
 default value of • CD-95
 scope of • CD-94
 type of • CD-95
 usage of • CD-94, CD-95
 creating • 6-4, CD-53
 example • 6-5
 creating a general template file using variables • 6-8
 defined by DEC/Test Manager • 4-6, 6-11
 DTM\$COLLECTION_NAME • 6-13
 DTM\$RESULT • 6-11
 DTM\$TEST_NAME • 6-12
 defined by the user • 6-3
 defining • CD-53
 scope • 6-4
 usage • 6-4

Variables (cont'd.)

deleting • 6-6, CD-69
 restrictions on • 6-6
description of • 3-3, 6-3
displaying information about • CD-189
global • 6-6
 assigning a value to
 restriction on • 6-6
 associating modified variable values with existing collections • 6-6
 description of • 6-3
 modifying • 6-6
 overriding the default value for a collection
 • 6-8
 example • 6-8
 uses of • 6-6
 when available for use • 6-3, 6-4
local • 6-7
 adding to a new test description • 6-7
 adding to an existing test description • 6-7
 adding to a test description • CD-91
 changing the value of • CD-91
 description of • 6-3
 removing from a test description • 6-7, CD-91
 uses of • 6-7
 when available for use • 6-3, 6-4
modifying • 6-5, CD-96
naming • 6-4
 restriction on • 6-4
numeric • 6-4
overriding the default value
 caution about • 6-8
overriding the default value for a test description • 6-7
 effects of • 6-7
 example • 6-8
removing
 from a test description • CD-69
 removing from a test description • 6-6, CD-91
 removing the remark from • CD-95
 replacing the remark • CD-95
string • 6-4
 uses for • 1-24 table
 uses of • 6-3
 using in a template file • 3-3
Variable-value • 6-4, CD-5

VAX Performance and Coverage Analyzer

See also PCA (Review subsystem command)

See Performance and Coverage Analyzer

VERIFY • CD-201

/LOG • CD-201

/NOLOG • CD-201

/RECOVER • 5-24, 6-31, 6-32, CD-201,
CD-202

extraneous subdirectories • CD-202

remark • CD-201

Verify

correcting library errors • CD-201

VERIFY/RECOVER

See VERIFY, /RECOVER

Verifying a library

See Library, verifying

Version of DEC/Test Manager

displaying • CD-191

W

Wildcard characters • CD-8

Asterisk (*) • CD-8

percent sign (%) • CD-8

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

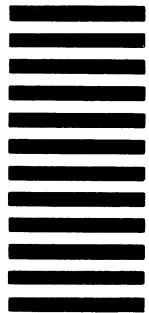
or Country

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here

Cut Along Dotted Line

**READER'S
COMMENTS**

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here

Cut Along Dotted Line