

HP ACMS for OpenVMS

Quick Reference Guide

Order Number: AA-LD80G-TE

January 2006

This manual contains quick reference information about the syntax of the utilities of *HP ACMS for OpenVMS*.

Revision/Update Information:	This manual supersedes the <i>HP ACMS for OpenVMS Quick Reference Guide, Version 4.5A</i> .
Operating System:	OpenVMS Alpha Version 8.2 OpenVMS I64 Version 8.2-1
Software Version:	<i>HP ACMS for OpenVMS, Version 5.0</i>

Hewlett-Packard Company
Palo Alto, California

© Copyright 2006 Hewlett-Packard Development Company, L.P.

Confidential Computer Software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation and Technical Data for Commercial Items are licenced to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained here in.

Motif is a registered US trademark of The Open Group.

Oracle is a registered US trademark of Oracle Corporation, Redwood City, California.

Oracle CODASYL DBMS, Oracle CDD/Administrator, Oracle CDD/Repository, Oracle Rdb, Oracle SQL/Services, Oracle Trace, and Oracle Trace Collector are registered US trademarks of Oracle Corporation, Redwood City, California.

Printed in US

Contents

Preface	xi
1 ADU Commands and Clauses	
1.1 ADU Commands	1-1
1.1.1 @ (At sign) Command (ADU>)	1-2
1.1.2 ATTACH Command (ADU>)	1-2
1.1.3 BUILD Command (ADU>)	1-2
1.1.4 COMPILE Command (ADU>)	1-3
1.1.5 COPY Command (ADU>)	1-3
1.1.6 CREATE Command (ADU>)	1-3
1.1.7 DELETE Command (ADU>)	1-4
1.1.8 DUMP Command (ADU>)	1-4
1.1.9 EDIT Command (ADU>)	1-4
1.1.10 EXIT Command (ADU>)	1-4
1.1.11 HELP Command (ADU>)	1-4
1.1.12 LINK Command (ADU>)	1-5
1.1.13 LIST Command (ADU>)	1-5
1.1.14 MODIFY Command (ADU>)	1-5
1.1.15 REPLACE Command (ADU>)	1-6
1.1.16 SAVE Command (ADU>)	1-6
1.1.17 SET DEFAULT Command (ADU>)	1-6
1.1.18 SET LOG Command (ADU>)	1-6
1.1.19 SET VERIFY Command (ADU>)	1-6
1.1.20 SHOW DEFAULT Command (ADU>)	1-7
1.1.21 SHOW LOG Command (ADU>)	1-7
1.1.22 SHOW VERSION Command (ADU>)	1-7
1.1.23 SPAWN Command (ADU>)	1-7
1.2 %INCLUDE	1-8
1.2.1 %INCLUDE	1-8
1.3 Task Definition Clauses	1-8
1.3.1 Task Syntax	1-9
1.3.2 Block Step Phrases Syntax	1-11
1.3.3 Exchange Step Syntax	1-12
1.3.4 Processing Step Syntax	1-14
1.3.5 Action Clauses Syntax	1-15
1.3.6 BLOCK Clause (Block)	1-16
1.3.7 CALL Clause (Processing)	1-16
1.3.8 CALL TASK Clause (Processing)	1-16
1.3.9 CANCEL ACTION Phrase (Block)	1-16
1.3.10 CANCEL TASK Clause (Action)	1-17
1.3.11 CANCELABLE Clause (Task)	1-17
1.3.12 COMMIT TRANSACTION Clause (Action)	1-17
1.3.13 CONTROL FIELD Clause (Action, Block, Exchange, Processing)	1-17

1.3.14	DATATRIEVE COMMAND Clause (Processing)	1-17
1.3.15	DCL COMMAND Clause (Processing)	1-17
1.3.16	DEFAULT FORM Clause (Task)	1-17
1.3.17	DEFAULT REQUEST LIBRARY Clause (Task)	1-18
1.3.18	DEFAULT SERVER Clause (Task)	1-18
1.3.19	DELAY Clause (Task)	1-18
1.3.20	EXCEPTION HANDLER Clause (Block, Exchange, Processing)	1-18
1.3.21	EXCHANGE Clause (Task)	1-18
1.3.22	EXIT BLOCK Clause (Action)	1-18
1.3.23	EXIT TASK Clause (Action)	1-19
1.3.24	FORM I/O Phrase (Block)	1-19
1.3.25	GET ERROR MESSAGE Clause (Action)	1-19
1.3.26	GLOBAL Clause (Task)	1-19
1.3.27	GOTO STEP Clause (Action)	1-19
1.3.28	IF THEN ELSE Clause (Action, Block, Exchange, Processing)	1-20
1.3.29	IMAGE Clause (Processing)	1-20
1.3.30	LOCAL Clause (Task)	1-20
1.3.31	MOVE Clause (Action)	1-20
1.3.32	NO EXCHANGE Clause (Exchange)	1-20
1.3.33	NO PROCESSING Clause (Processing)	1-20
1.3.34	NO SERVER CONTEXT ACTION Clause (Action)	1-20
1.3.35	NO TERMINAL I/O Phrase (Block, Processing)	1-21
1.3.36	NONPARTICIPATING SERVER Phrase (Processing)	1-21
1.3.37	PROCESSING Clause (Task)	1-21
1.3.38	RAISE EXCEPTION Clause (Action)	1-21
1.3.39	READ Clause (Exchange)	1-21
1.3.40	RECEIVE Clause (Exchange)	1-22
1.3.41	RELEASE SERVER CONTEXT Clause (Action)	1-22
1.3.42	REPEAT STEP Clause (Action)	1-22
1.3.43	REQUEST Clause (Exchange)	1-22
1.3.44	REQUEST I/O Phrase (Block, Processing)	1-22
1.3.45	RETAIN SERVER CONTEXT Clause (Action)	1-22
1.3.46	ROLLBACK TRANSACTION Clause (Action)	1-23
1.3.47	SELECT FIRST Clause (Action, Block, Exchange, Processing)	1-23
1.3.48	SEND Clause (Exchange)	1-23
1.3.49	SERVER CONTEXT Phrase (Block)	1-23
1.3.50	STREAM I/O Phrase (Block)	1-23
1.3.51	TASK ARGUMENTS Phrase (Task)	1-24
1.3.52	TERMINAL I/O Phrase (Processing)	1-24
1.3.53	TRANSACTION Phrase (Block, Processing)	1-24
1.3.54	TRANSCEIVE Clause (Exchange)	1-24
1.3.55	USE WORKSPACE Clause (Task)	1-25
1.3.56	WAIT Clause (Task)	1-25
1.3.57	WHILE DO Clause (Block, Exchange, Processing)	1-25
1.3.58	WORKSPACES Clause (Task)	1-25
1.3.59	WRITE Clause (Exchange)	1-26
1.4	Task Group Definition Clauses	1-26
1.4.1	Task Group Syntax	1-27
1.4.2	Processing Subclauses Syntax	1-28
1.4.3	Server Subclauses Syntax	1-29
1.4.4	ALWAYS EXECUTE TERMINATION PROCEDURE Subclause (Server)	1-30
1.4.5	CALL Subclause (Processing)	1-30
1.4.6	CANCEL PROCEDURE Subclause (Server)	1-30

1.4.7	DATATRIEVE COMMAND Subclause (Processing)	1-30
1.4.8	DCL AVAILABLE Subclause (Server)	1-30
1.4.9	DCL COMMAND Subclause (Processing)	1-30
1.4.10	DCL PROCESS Subclause (Server)	1-30
1.4.11	DEFAULT OBJECT FILE Subclause (Server)	1-31
1.4.12	DEFAULT TASK GROUP FILE Clause (Task Group)	1-31
1.4.13	DYNAMIC USERNAME Subclause (Server)	1-31
1.4.14	FIXED USERNAME Subclause (Server)	1-31
1.4.15	FORMS Clause (Task Group)	1-31
1.4.16	IMAGE Subclause (Processing)	1-31
1.4.17	INITIALIZATION PROCEDURE Subclause (Server)	1-31
1.4.18	MESSAGE FILES Clause (Task Group)	1-32
1.4.19	PROCEDURE SERVER IMAGE Subclause (Server)	1-32
1.4.20	PROCEDURES Subclause (Server)	1-32
1.4.21	REQUEST LIBRARIES Clause (Task Group)	1-32
1.4.22	REUSABLE Subclause (Server)	1-32
1.4.23	RUNDOWN ON CANCEL Subclause (Server)	1-32
1.4.24	SERVERS Clause (Task Group)	1-33
1.4.25	TASKS Clause (Task Group)	1-33
1.4.26	TERMINATION PROCEDURE Subclause (Server)	1-33
1.4.27	USERNAME Subclause (Server)	1-33
1.4.28	WORKSPACES Clause (Task Group)	1-34
1.5	Application Definition Clauses	1-34
1.5.1	Application Definition Syntax	1-35
1.5.2	SERVER ATTRIBUTES Clause Syntax	1-37
1.5.3	SERVER DEFAULTS Clause Syntax	1-38
1.5.4	TASK ATTRIBUTES Clause Syntax	1-40
1.5.5	TASK DEFAULTS Clause Syntax	1-41
1.5.6	ACCESS Subclause (Task)	1-42
1.5.7	APPLICATION DEFAULT DIRECTORY Clause (Application)	1-42
1.5.8	APPLICATION LOGICALS Clause (Application)	1-42
1.5.9	APPLICATION NAME TABLES Clause (Application)	1-42
1.5.10	APPLICATION USERNAME Clause (Application)	1-42
1.5.11	AUDIT Clause (Application, Server, Task)	1-42
1.5.12	CANCELABLE Subclause (Task)	1-43
1.5.13	CREATION DELAY Subclause (Server)	1-43
1.5.14	CREATION INTERVAL Subclause (Server)	1-43
1.5.15	DEFAULT APPLICATION FILE Clause (Application)	1-43
1.5.16	DEFAULT DIRECTORY Subclause (Server)	1-43
1.5.17	DELAY Subclause (Task)	1-43
1.5.18	DELETION DELAY Subclause (Server)	1-43
1.5.19	DELETION INTERVAL Subclause (Server)	1-44
1.5.20	DISABLE Subclause (Task)	1-44
1.5.21	DYNAMIC USERNAME Subclause (Server)	1-44
1.5.22	ENABLE Subclause (Task)	1-44
1.5.23	FIXED USERNAME Subclause (Server)	1-44
1.5.24	GLOBAL Subclause (Task)	1-44
1.5.25	LOCAL Subclause (Task)	1-44
1.5.26	LOGICALS Subclause (Server)	1-45
1.5.27	MAXIMUM SERVER PROCESSES Clause (Application, Server)	1-45
1.5.28	MAXIMUM TASK INSTANCES Clause (Application)	1-45
1.5.29	MINIMUM SERVER PROCESSES Subclause (Server)	1-45
1.5.30	NAME TABLES Subclause (Server)	1-45
1.5.31	PROTECTED WORKSPACES Subclause (Server)	1-46

1.5.32	SERVER ATTRIBUTES Clause (Application)	1-46
1.5.33	SERVER DEFAULTS Clause (Application)	1-46
1.5.34	SERVER MONITORING INTERVAL Clause (Application)	1-46
1.5.35	SERVER PROCESS DUMP Subclause (Server)	1-46
1.5.36	TASK ATTRIBUTES Clause (Application)	1-47
1.5.37	TASK DEFAULTS Clause (Application)	1-47
1.5.38	TASK GROUPS Clause (Application)	1-47
1.5.39	TRANSACTION TIMEOUT Subclause (Task)	1-47
1.5.40	USERNAME Subclause (Server)	1-47
1.5.41	WAIT Subclause (Task)	1-48
1.6	Menu Definition Clauses	1-48
1.6.1	Menu Definition Syntax	1-49
1.6.2	CONTROL TEXT Clause (Menu)	1-50
1.6.3	DEFAULT APPLICATION Clause (Menu)	1-50
1.6.4	DEFAULT MENU FILE Clause (Menu)	1-50
1.6.5	DELAY Subclause (Optional ENTRIES)	1-50
1.6.6	ENTRIES Clause (Menu)	1-50
1.6.7	HEADER Clause (Menu)	1-50
1.6.8	MENU Subclause (Required ENTRIES)	1-51
1.6.9	REQUEST Clause (Menu)	1-51
1.6.10	TASK Subclause (Required ENTRIES)	1-51
1.6.11	TEXT Subclause (Optional ENTRIES)	1-51
1.6.12	WAIT Subclause (Optional ENTRIES)	1-51
1.6.13	Application Specification Parameter	1-51
1.7	Declining Features Syntax	1-52
1.7.1	COMMIT Clause (Action)	1-52
1.7.2	CONTINUE ON BAD STATUS Phrase (Processing)	1-52
1.7.3	DBMS RECOVERY Phrase (Block, Processing)	1-52
1.7.4	GOTO TASK Clause (Action)	1-52
1.7.5	NO RECOVERY UNIT ACTION Clause (Action)	1-52
1.7.6	RDB RECOVERY Phrase (Block, Processing)	1-53
1.7.7	REPEAT TASK Clause (Action)	1-53
1.7.8	RETAIN RECOVERY UNIT Clause (Action)	1-53
1.7.9	RMS RECOVERY Phrase (Block, Processing)	1-53
1.7.10	ROLLBACK Clause (Action)	1-53
1.7.11	SQL RECOVERY Phrase (Block, Processing)	1-53

2 ACMS Management Utilities and Commands

2.1	ACMSQUEMGR Commands	2-1
2.1.1	CREATE QUEUE Command (ACMSQUEMGR>)	2-1
2.1.2	DELETE ELEMENT Command (ACMSQUEMGR>)	2-1
2.1.3	DELETE QUEUE Command (ACMSQUEMGR>)	2-1
2.1.4	EXIT Command (ACMSQUEMGR>)	2-2
2.1.5	HELP Command (ACMSQUEMGR>)	2-2
2.1.6	MODIFY QUEUE Command (ACMSQUEMGR>)	2-2
2.1.7	SET ELEMENT Command (ACMSQUEMGR>)	2-2
2.1.8	SET QUEUE Command (ACMSQUEMGR>)	2-2
2.1.9	SHOW ELEMENT Command (ACMSQUEMGR>)	2-3
2.1.10	SHOW QUEUE Command (ACMSQUEMGR>)	2-3
2.2	AAU Commands	2-3
2.2.1	ADD Command (AAU>)	2-3
2.2.2	COPY Command (AAU>)	2-3
2.2.3	DEFAULT Command (AAU>)	2-4

2.2.4	EXIT Command (AAU>)	2-4
2.2.5	HELP Command (AAU>)	2-4
2.2.6	LIST Command (AAU>)	2-4
2.2.7	MODIFY Command (AAU>)	2-4
2.2.8	REMOVE Command (AAU>)	2-5
2.2.9	RENAME Command (AAU>)	2-5
2.2.10	SHOW Command (AAU>)	2-5
2.3	ACMSGEN Commands	2-5
2.3.1	EXIT Command (ACMSGEN>)	2-5
2.3.2	HELP Command (ACMSGEN>)	2-5
2.3.3	SET Command (ACMSGEN>)	2-6
2.3.4	SHOW Command (ACMSGEN>)	2-6
2.3.5	USE Command (ACMSGEN>)	2-6
2.3.6	USE ACTIVE Command (ACMSGEN>)	2-6
2.3.7	USE CURRENT Command (ACMSGEN>)	2-6
2.3.8	USE DEFAULT Command (ACMSGEN>)	2-6
2.3.9	WRITE Command (ACMSGEN>)	2-7
2.3.10	WRITE ACTIVE Command (ACMSGEN>)	2-7
2.3.11	WRITE CURRENT Command (ACMSGEN>)	2-7
2.4	ATR Commands	2-7
2.4.1	EXIT Command (ATR>)	2-7
2.4.2	HELP Command (ATR>)	2-7
2.4.3	LIST Command (ATR>)	2-7
2.5	DDU Commands	2-8
2.5.1	ADD Command (DDU>)	2-8
2.5.2	COPY Command (DDU>)	2-8
2.5.3	DEFAULT Command (DDU>)	2-8
2.5.4	EXIT Command (DDU>)	2-9
2.5.5	HELP Command (DDU>)	2-9
2.5.6	LIST Command (DDU>)	2-9
2.5.7	MODIFY Command (DDU>)	2-9
2.5.8	REMOVE Command (DDU>)	2-9
2.5.9	RENAME Command (DDU>)	2-9
2.5.10	SHOW Command (DDU>)	2-10
2.6	Operator Commands	2-10
2.6.1	ACMS/CANCEL TASK Command	2-10
2.6.2	ACMS/CANCEL USER Command	2-10
2.6.3	ACMS/DEBUG Command	2-11
2.6.4	ACMS/ENTER Command	2-11
2.6.5	ACMS/INSTALL Command	2-11
2.6.6	ACMS/MODIFY APPLICATION Command	2-11
2.6.7	ACMS/REPLACE SERVER Command	2-12
2.6.8	ACMS/REPROCESS APPLICATION_SPEC Command	2-12
2.6.9	ACMS/RESET AUDIT Command	2-12
2.6.10	ACMS/RESET TERMINALS Command	2-12
2.6.11	ACMS/SET QUEUE Command	2-12
2.6.12	ACMS/SET SYSTEM Command	2-13
2.6.13	ACMS/SHOW APPLICATION Command	2-13
2.6.14	ACMS/SHOW APPLICATION/CONTINUOUS Command	2-13
2.6.15	ACMS/SHOW QTI Command	2-13
2.6.16	ACMS/SHOW QUEUE Command	2-13
2.6.17	ACMS/SHOW SERVER Command	2-14
2.6.18	ACMS/SHOW SYSTEM Command	2-14
2.6.19	ACMS/SHOW TASK Command	2-14

2.6.20	ACMS/SHOW USER Command	2-14
2.6.21	ACMS/START APPLICATION Command	2-15
2.6.22	ACMS/START QTI Command	2-15
2.6.23	ACMS/START QUEUE Command	2-15
2.6.24	ACMS/START SYSTEM Command	2-15
2.6.25	ACMS/START TASK Command	2-15
2.6.26	ACMS/START TERMINALS Command	2-15
2.6.27	ACMS/STOP APPLICATION Command	2-16
2.6.28	ACMS/STOP QTI Command	2-16
2.6.29	ACMS/STOP QUEUE Command	2-16
2.6.30	ACMS/STOP SYSTEM Command	2-16
2.6.31	ACMS/STOP TERMINALS Command	2-16
2.7	SWLUP Commands	2-16
2.7.1	@ (At sign) Command (SWLUP>)	2-17
2.7.2	EDIT Command (SWLUP>)	2-17
2.7.3	EXIT Command (SWLUP>)	2-17
2.7.4	HELP Command (SWLUP>)	2-17
2.7.5	LIST Command (SWLUP>)	2-17
2.7.6	RENEW Command (SWLUP>)	2-17
2.7.7	SAVE Command (SWLUP>)	2-18
2.7.8	SET [NO]LOG Command (SWLUP>)	2-18
2.7.9	SET [NO]VERIFY Command (SWLUP>)	2-18
2.7.10	SHOW CURRENT Command (SWLUP>)	2-18
2.7.11	SHOW LOG Command (SWLUP>)	2-18
2.7.12	SHOW VERSION Command (SWLUP>)	2-18
2.7.13	STOP Command (SWLUP>)	2-18
2.8	UDU Commands	2-19
2.8.1	ADD Command (UDU>)	2-19
2.8.2	ADD/PROXY Command (UDU>)	2-19
2.8.3	COPY Command (UDU>)	2-19
2.8.4	CREATE/PROXY Command (UDU>)	2-20
2.8.5	DEFAULT Command (UDU>)	2-20
2.8.6	EXIT Command (UDU>)	2-20
2.8.7	HELP Command (UDU>)	2-20
2.8.8	LIST Command (UDU>)	2-21
2.8.9	LIST/PROXY Command (UDU>)	2-21
2.8.10	MODIFY Command (UDU>)	2-21
2.8.11	REMOVE Command (UDU>)	2-21
2.8.12	REMOVE/PROXY Command (UDU>)	2-21
2.8.13	RENAME Command (UDU>)	2-22
2.8.14	SHOW Command (UDU>)	2-22
2.8.15	SHOW/PROXY Command (UDU>)	2-22

3 ACMS Application Programming Services and Task Debugger Commands

3.1	ACMS Application Programming Services	3-1
3.1.1	ACMS\$GET_TID	3-1
3.1.2	ACMS\$RAISE_NONREC_EXCEPTION	3-1
3.1.3	ACMS\$RAISE_STEP_EXCEPTION	3-1
3.1.4	ACMS\$RAISE_TRANS_EXCEPTION	3-1
3.1.5	ACMSAD\$REQ_CANCEL	3-2
3.1.6	ACMS\$DEQUEUE_TASK	3-2
3.1.7	ACMS\$QUEUE_TASK	3-2

3.2	ACMS Task Debugger Commands	3-3
3.2.1	@ (At sign) Command	3-3
3.2.2	ACCEPT Command	3-3
3.2.3	ASSIGN Command	3-3
3.2.4	CANCEL BREAK Command	3-3
3.2.5	CANCEL TASK Command	3-3
3.2.6	CANCEL TRANSACTION_TIMEOUT Command	3-4
3.2.7	DEPOSIT Command	3-4
3.2.8	EXAMINE Command	3-4
3.2.9	EXIT Command	3-4
3.2.10	GO Command	3-4
3.2.11	HELP Command	3-4
3.2.12	INTERRUPT Command	3-5
3.2.13	SELECT Command	3-5
3.2.14	SET BREAK Command	3-5
3.2.15	SET SERVER Command	3-5
3.2.16	SET TRANSACTION_TIMEOUT	3-5
3.2.17	SHOW BREAK Command	3-5
3.2.18	SHOW SERVERS Command	3-5
3.2.19	SHOW TRANSACTION_TIMEOUT	3-6
3.2.20	SHOW VERSION Command	3-6
3.2.21	START Command	3-6
3.2.22	STEP Command	3-6
3.2.23	STOP Command	3-6

4 Systems Interface (SI) Services

4.1	Initialization and Exchange I/O Services	4-1
4.1.1	ACMS\$INIT_EXCHANGE_IO	4-1
4.1.2	ACMS\$SIGN_IN	4-1
4.1.3	ACMS\$SIGN_OUT	4-2
4.1.4	ACMS\$TERM_EXCHANGE_IO	4-2
4.2	Submitter Services	4-2
4.2.1	ACMS\$CALL	4-2
4.2.2	ACMS\$CANCEL_CALL	4-3
4.2.3	ACMS\$GET_PROCEDURE_INFO	4-3
4.2.4	ACMS\$START_CALL	4-3
4.2.5	ACMS\$WAIT_FOR_CALL_END	4-4
4.3	Stream Services	4-4
4.3.1	ACMS\$REPLY_TO_STREAM_IO	4-4
4.3.2	ACMS\$WAIT_FOR_STREAM_IO	4-4
4.4	Superseded Services	4-5
4.4.1	ACMS\$CLOSE_RR	4-5
4.4.2	ACMS\$CONNECT_STREAM	4-5
4.4.3	ACMS\$CREATE_STREAM	4-6
4.4.4	ACMS\$DELETE_STREAM	4-6
4.4.5	ACMS\$DISCONNECT_STREAM	4-6
4.4.6	ACMS\$OPEN_RR	4-6

A Checklist for ACMS Application Development

B Changing and Debugging ACMS Applications

C Summary of ACMS System Workspaces

C.1	ACMS\$PROCESSING_STATUS System Workspace	C-1
C.2	ACMS\$SELECTION_STRING System Workspace	C-2
C.3	ACMS\$TASK_INFORMATION System Workspace	C-2

Figures

A-1	ACMS Application Design, Development, and Use	A-1
-----	---	-----

Tables

1-1	Startup Qualifiers and Their Functions	1-1
B-1	Changing ACMS Applications	B-1
B-2	Files Used to Debug ACMS Tasks	B-4
B-3	Source Files for Debugging	B-5
C-1	Fields in ACMS\$PROCESSING_STATUS	C-1
C-2	Fields in ACMS\$SELECTION_STRING	C-2
C-3	Fields in ACMS\$TASK_INFORMATION	C-2

Preface

Intended Audience

This manual is a quick reference to all the syntax for the utilities of *HP ACMS for OpenVMS* (ACMS) software.

This manual is intended for application designers, programmers, and anyone responsible for managing applications using ACMS.

Document Structure

This manual has four chapters and three appendixes:

Chapter 1	Lists syntax for the ACMS Application Definition Utility (ADU) commands and clauses.
Chapter 2	Lists syntax for the ACMS management utilities and operator commands.
Chapter 3	Lists syntax for the ACMS programming services and Task Debugger commands.
Chapter 4	Lists syntax for the ACMS Systems Interface services.
Appendix A	Illustrates and provides a detailed checklist of the phases of ACMS application development.
Appendix B	Summarizes the relationships between different parts of ACMS applications.
Appendix C	Describes ACMS system workspaces.

ACMS Help

ACMS and its components provide extensive online help.

- DCL level help

Enter `HELP ACMS` at the DCL prompt for complete help about the ACMS command and qualifiers, and for other elements of ACMS for which independent help systems do not exist. DCL level help also provides brief help messages for elements of ACMS that contain independent help systems (such as the ACMS utilities) and for related products used by ACMS (such as DECforms or Oracle CDD/Repository).

- ACMS utilities help

Each of the following ACMS utilities has an online help system:

- ACMS Debugger
- ACMSGEN Utility
- ACMS Queue Manager (ACMSQUEMGR)
- Application Definition Utility (ADU)
- Application Authorization Utility (AAU)
- Device Definition Utility (DDU)
- User Definition Utility (UDU)
- Audit Trail Report Utility (ATR)
- Software Event Log Utility Program (SWLUP)

The two ways to get utility-specific help are:

- Run the utility and type HELP at the utility prompt.
- Use the DCL HELP command. At the “Topic?” prompt, type @ followed by the name of the utility. Use the ACMS prefix, even if the utility does not have an ACMS prefix (except for SWLUP). For example:

```
Topic? @ACMSQUEMGR
Topic? @ACMSADU
```

However, do not use the ACMS prefix with SWLUP:

```
Topic? @SWLUP
```

Note that if you run the ACMS Debugger Utility and then type HELP, you must specify a file. If you ask for help from the DCL level with @, you do not need to specify a file.

- ACMSPARAM.COM and ACMEXCPAR.COM help

Help for the command procedures that set parameters and quotas is a subset of the DCL level help. You have access to this help from the DCL prompt, or from within the command procedures.

- LSE help

ACMS provides ACMS-specific help within the LSE templates that assist in the creation of applications, tasks, task groups, and menus. The ACMS-specific LSE help is a subset of the ADU help system. Within the LSE templates, this help is context-sensitive. Type HELP/IND (PF1-PF2) at any placeholder for which you want help.

- Error help

ACMS and each of its utilities provide error message help. Use HELP ACMS ERRORS from the DCL prompt for ACMS error message help. Use HELP ERRORS from the individual utility prompts for error message help for that utility.

- Terminal user help

At each menu within an ACMS application, ACMS provides help about terminal user commands, special key mappings, and general information about menus and how to select tasks from menus.

- Forms help

For complete help for HP DECforms or HP TDMS, use the help systems for these products.

Related Documents

The following table lists the books in the *HP ACMS for OpenVMS* documentation set.

ACMS Information	Description
<i>HP ACMS Version 5.0 for OpenVMS Release Notes</i> †	Information about the latest release of the software
<i>HP ACMS Version 5.0 for OpenVMS Installation Guide</i>	Description of installation requirements, the installation procedure, and postinstallation tasks.
<i>HP ACMS for OpenVMS Getting Started</i>	Overview of ACMS software and documentation. Tutorial for developing a simple ACMS application. Description of the AVERTZ sample application.
<i>HP ACMS for OpenVMS Concepts and Design Guidelines</i>	Description of how to design an ACMS application.
<i>HP ACMS for OpenVMS Writing Applications</i>	Description of how to write task, task group, application, and menu definitions using the Application Definition Utility. Description of how to write and migrate ACMS applications on an OpenVMS Alpha system.
<i>HP ACMS for OpenVMS Writing Server Procedures</i>	Description of how to write programs to use with tasks and how to debug tasks and programs. Description of how ACMS works with the APPC/LU6.2 programming interface to communicate with IBM CICS applications. Description of how ACMS works with third-party database managers, with Oracle used as an example.
<i>HP ACMS for OpenVMS Systems Interface Programming</i>	Description of using Systems Interface (SI) Services to submit tasks to an ACMS system.
<i>HP ACMS for OpenVMS ADU Reference Manual</i>	Reference information about the ADU commands, phrases, and clauses.
<i>HP ACMS for OpenVMS Quick Reference</i>	List of ACMS syntax with brief descriptions.
<i>HP ACMS for OpenVMS Managing Applications</i>	Description of authorizing, running, and managing ACMS applications, and controlling the ACMS system.
<i>HP ACMS for OpenVMS Remote Systems Management Guide</i>	Description of the features of the Remote Manager for managing ACMS systems, how to use the features, and how to manage the Remote Manager.
Online help†	Online help about ACMS and its utilities.

†Available on line only.

For additional information on the compatibility of other software products with this version of ACMS, refer to the *HP ACMS for OpenVMS Software Product Description* (SPD 25.50.xx).

For additional information about the Open Systems Software Group (OSSG) products and services, access the following OpenVMS World Wide Web address:

<http://h71000.www7.hp.com/openvms>

Reader's Comments

HP welcomes your comments on this manual.

Print or edit the online form `SYS$HELP:OPENVMSDOC_COMMENTS.TXT` and send us your comments by:

Internet	openvmsdoc@hp.com
Mail	Hewlett-Packard Company, L.P. OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following World Wide Web address for information about how to order additional documentation:

<http://www.hp.com/go/openvms/doc/>

To reach the OpenVMS documentation website, click the Documentation link.

If you need help deciding which documentation best meets your needs, call 1-800-ATCOMPA.

Conventions

The following conventions are used in this manual:

<code>Ctrl/x</code>	A sequence such as <code>Ctrl/x</code> indicates that you must press and hold the key labeled <code>Ctrl</code> while you press another key or a pointing device button.
<code>PF1 x</code>	A sequence such as <code>PF1 x</code> indicates that you must first press and release the key labeled <code>PF1</code> and then press and release another key or a pointing device button.
<code>Return</code>	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.) In the HTML version of this document, this convention appears as brackets rather than a box.
<code>...</code>	A horizontal ellipsis in examples indicates one of the following possibilities: <ul style="list-style-type: none">• Additional optional arguments in a statement have been omitted.• The preceding item or items can be repeated one or more times.• Additional parameters, values, or other information can be entered.
<code>.</code>	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.

Monospace text	<p>Monospace type indicates code examples and interactive screen displays.</p> <p>In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.</p> <p>In the HTML version of this document, this text style may appear as italics.</p>
-	<p>A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.</p>
numbers	<p>All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.</p>
bold text	<p>Bold text represents the introduction of a new term or the name of an argument, an attribute, or a reason.</p> <p>In the HTML version of this document, this text style may appear as italics.</p>
<i>italic text</i>	<p>Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i>), in command lines (<i>/PRODUCER=name</i>), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).</p>
UPPERCASE	<p>Uppercase text indicates the name of a routine, the name of a file, the name of a file protection code, or the abbreviation for a system privilege.</p> <p>In command format descriptions, uppercase text is an optional keyword.</p>
<u>UPPERCASE</u>	<p>In command format descriptions, uppercase text that is underlined is required. You must include it in the statement if the clause is used.</p>
lowercase	<p>In command format descriptions, a lowercase word indicates a required element.</p>
<lowercase>	<p>In command format descriptions, lowercase text in angle brackets indicates a required clause or phrase.</p>
()	<p>In command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.</p>
[]	<p>In command format descriptions, vertical bars within square brackets indicate that you can choose any combination of the enclosed options, but you can choose each option only once.</p>
{ }	<p>In command format descriptions, vertical bars within braces indicate that you must choose one of the options listed, but you can use each option only once.</p>

References to Products

The ACMS documentation set to which this manual belongs often refers to certain products by abbreviated names:

Abbreviation	Product
ACMS	<i>HP ACMS for OpenVMS Alpha, and HP ACMS for OpenVMS I64</i>
Ada	HP Ada for OpenVMS Alpha Systems, and HP Ada for OpenVMS I64 Systems
BASIC	HP BASIC for OpenVMS
C	HP C for OpenVMS Alpha Systems, and HP C for OpenVMS I64 Systems
CDD	Oracle CDD/Administrator, and Oracle CDD/Repository
COBOL	HP COBOL for OpenVMS Alpha Systems, and HP COBOL for OpenVMS I64 Systems
DATATRIEVE	HP DATATRIEVE for OpenVMS Alpha, and HP DATATRIEVE for OpenVMS I64
DBMS	Oracle CODASYL DBMS
DECforms	HP DECforms
FORTRAN	HP Fortran for OpenVMS Alpha Systems, and HP Fortran for OpenVMS I64 Systems
OpenVMS	The OpenVMS Alpha operating system, and the OpenVMS I64 operating system
Pascal	HP Pascal for OpenVMS Alpha, and HP Pascal for OpenVMS I64
Rdb	Oracle Rdb
SQL	The SQL interface to Oracle Rdb

ADU Commands and Clauses

This chapter contains syntax for the commands and clauses of the ACMS Application Definition Utility (ADU). ADU commands allow you to create or change definitions for ACMS tasks, task groups, applications, and menus. The definitions themselves are made up of ADU clauses and phrases. See the *HP ACMS for OpenVMS ADU Reference Manual* for more information about ADU commands and clauses.

1.1 ADU Commands

Use ADU commands to create or change the definitions an ACMS application uses, or to gather information about your own work or that of an ACMS application. You can issue ADU commands interactively or in a command file.

To create an ACMS application, use ADU commands to reset your data dictionary default directory during a session, as well as write, change, copy, delete, and compile definitions for tasks, task groups, menus, and applications. Use other ADU commands to build or rebuild task group, menu, and application database files.

To gather information about your work, you can use ADU commands to check the version of ADU on your system, log an interactive session to a file in your default directory for later reference, check if logging is active, and verify the work a command file performs during execution. To gather information about an ACMS application, you can use ADU commands to list the contents of task group, application, or menu database files, so you can check the consistency of procedure names, workspaces, and servers. See the *HP ACMS for OpenVMS ADU Reference Manual* for a detailed explanation of the syntax of ADU commands.

Table 1–1 lists the ADU startup command qualifiers.

Table 1–1 Startup Qualifiers and Their Functions

Qualifier	Function
/COMMAND [=file-spec] /NOCOMMAND	Tells ADU whether or not to execute a startup command file when you invoke the utility. By default, when you invoke ADU, it runs a command file named ADUINI.COM, located in your default directory. To invoke a different startup command file, include its file specification with the /COMMAND qualifier. When you specify the /NOCOMMAND qualifier, ACMS starts the ADU without executing any startup command file.

(continued on next page)

ADU Commands and Clauses

1.1 ADU Commands

Table 1–1 (Cont.) Startup Qualifiers and Their Functions

Qualifier	Function
/JOURNAL /NOJOURNAL	By default, ADU creates a journal file that contains every keystroke made during your ADU session. The journal file, named ADUJNL.JOU, is located in your default directory. The journal file is saved if your ADU session is interrupted. When you exit normally (by using the EXIT command or entering Ctrl/Z), the journal file is not saved. Use the /NOJOURNAL qualifier to turn off the journaling feature.
/PATH=path-name	Assigns a CDD directory. If you do not specify a path name, ADU uses the default CDD directory.
/RECOVER /NORECOVER	If you specify the /RECOVER qualifier, ADU runs the journal file, ADLJNL.JOU, to restore an ADU session that has ended abnormally. With /RECOVER in effect, ADU replays the interrupted session to recover your work. /NORECOVER is the default.

1.1.1 @ (At sign) Command (ADU>)

Executes a command file containing either ADU commands or ACMS definitions. If you do not specify a file type, ACMS supplies .COM as the default.

@ command-file-spec

1.1.2 ATTACH Command (ADU>)

Transfers control from your process to another process in your job.

ATTACH process-name

1.1.3 BUILD Command (ADU>)

Converts object definitions from the dictionary into binary database files that ACMS uses at run time.

BUILD { $\left. \begin{array}{l} \text{APPLICATION} \\ \text{GROUP} \\ \text{MENU} \end{array} \right\}$ path-name [database-file-spec] [/qualifiers]

Command Qualifiers

/AUDIT[=audit-list]
/NOAUDIT
/[NO]DEBUG
/LIST[=list-file-spec]
/NOLIST
/[NO]LOG
/OBJECT=(file-spec [...])
/NOOBJECT
/[NO]PRINT
/[NO]STD L
/[NO]SYSLIB
/[NO]SYSSHR
/USERLIBRARY=(file-spec [...])
/NOUSERLIBRARY

Defaults

/AUDIT=standard-audit-string
/NODEBUG
/LIST (Batch)
/NOLIST (Interactive)
/NOLOG
/NOOBJECT
/NOPRINT
/STD L
/SYSLIB
/SYSSHR
/NOUSERLIBRARY

1.1.4 COMPILE Command (ADU>)

Checks an application, task group, menu, or task definition for syntax errors, and writes the compilation results to a file.

```

COMPILE { APPLICATION
            GROUP
            MENU
          } reference-name [ definition-file-spec ] [ /qualifiers ]
  
```

Command Qualifiers	Defaults
/DIAGNOSTICS[=diagnostics-file-spec]	
/NODIAGNOSTICS	/NODIAGNOSTICS
/LIST[=list-file-spec]	/LIST (Batch)
/NOLIST	/NOLIST (Interactive)
/[NO]LOG	/NOLOG
/OUTPUT=output-file-spec [...]	
/[NO]PRINT	/NOPRINT

1.1.5 COPY Command (ADU>)

Creates a copy of a definition.

```

COPY { APPLICATION
          GROUP
          MENU
          TASK
        } src-path-name dst-path-name [ /qualifiers ]
  
```

Command Qualifiers	Defaults
/AUDIT[=audit-list]	/AUDIT=standard-audit-string
/NOAUDIT	
/[NO]LOG	/NOLOG

1.1.6 CREATE Command (ADU>)

Checks an application, task group, menu, or task definition for syntax errors, and stores valid new definitions in the dictionary.

```

CREATE { APPLICATION
            GROUP
            MENU
            TASK
          } path-name [file-spec] [/qualifiers]
  
```

Command Qualifiers	Defaults
/AUDIT[=audit-list]	/AUDIT=standard-audit-string
/NOAUDIT	
/DIAGNOSTICS[=diagnostics-file-spec]	
/NODIAGNOSTICS	/NODIAGNOSTICS
/LIST[=list-file-spec]	/LIST (Batch)
/NOLIST	/NOLIST (Interactive)
/[NO]LOG	/NOLOG
/[NO]PRINT	/NOPRINT

ADU Commands and Clauses

1.1 ADU Commands

1.1.7 DELETE Command (ADU>)

Removes a definition from the dictionary.

DELETE { APPLICATION
GROUP
MENU
TASK } path-name [/qualifiers]

Command Qualifiers

/[NO]CONFIRM
/[NO]LOG

Defaults

/NOCONFIRM
/NOLOG

1.1.8 DUMP Command (ADU>)

Displays the contents of an application, menu, or task group database file.

DUMP { APPLICATION
GROUP
MENU } database-file-spec [/qualifiers]

Command Qualifiers

/OUTPUT[=file-spec]
/NOOUTPUT
/[NO]PRINT

Defaults

/NOOUTPUT
/NOPRINT

1.1.9 EDIT Command (ADU>)

Invokes a text editor to let you make changes to the last command you entered.

EDIT

1.1.10 EXIT Command (ADU>)

Ends the current ADU session and returns you to the DCL prompt.

EXIT

1.1.11 HELP Command (ADU>)

Displays information about ADU commands and clauses.

HELP [/qualifier] [topic]

Command Qualifiers

/[NO]PROMPT

Defaults

/PROMPT

1.1.12 LINK Command (ADU>)

Converts object definitions from OpenVMS files into binary database files that ACMS uses at run time.

LINK { APPLICATION
GROUP
MENU } compile-result-file-spec [database-file-spec] [/qualifiers]

Command Qualifiers	Defaults
/AUDIT[=audit-list]	/AUDIT=standard-audit-string
/NOAUDIT	
/[NO]DEBUG	/NODEBUG
/LIST[=list-file-spec]	/LIST (Batch)
/NOLIST	/NOLIST (Interactive)
/[NO]LOG	/NOLOG
/OBJECT=(file-spec [...])	
/NOOBJECT	/NOOBJECT
/[NO]PRINT	/NOPRINT
/[NO]STDL	/STDL
/[NO]SYSLIB	/SYSLIB
/[NO]SYSSHR	/SYSSHR
/USERLIBRARY=(file-spec [...])	
/NOUSERLIBRARY	/NOUSERLIBRARY

1.1.13 LIST Command (ADU>)

Displays the contents of a definition in a dictionary directory.

LIST { APPLICATION
GROUP
MENU
TASK } path-name [/qualifiers]

Command Qualifiers	Defaults
/OUTPUT[=list-file-spec]	
/NOOUTPUT	/NOOUTPUT
/[NO]PRINT	/NOPRINT

1.1.14 MODIFY Command (ADU>)

Retrieves a definition from the dictionary and runs a text editor so you can change the definition.

MODIFY { APPLICATION
GROUP
MENU
TASK } path-name [/qualifiers]

Command Qualifiers	Defaults
/AUDIT[=audit-list]	/AUDIT=standard-audit-string
/NOAUDIT	
/DIAGNOSTICS[=diagnostics-file-spec]	
/NODIAGNOSTICS	/NODIAGNOSTICS
/LIST[=list-file-spec]	/NOLIST (Interactive)
/NOLIST	/LIST (Batch)

ADU Commands and Clauses

1.1 ADU Commands

/[NO]LOG
/[NO]PRINT

/NOLOG
/NOPRINT

1.1.15 REPLACE Command (ADU>)

Replaces an old dictionary definition with a new one.

```
REPLACE { APPLICATION  
         GROUP  
         MENU  
         TASK } path-name [file-spec] [/qualifiers]
```

Command Qualifiers

/AUDIT[=audit-list]
/NOAUDIT
/[NO]CREATE
/DIAGNOSTICS[=diagnostics-file-spec]
/NODIAGNOSTICS
/LIST[=list-file-spec]
/NOLIST
/[NO]LOG
/[NO]PRINT

Defaults

/AUDIT=standard-audit-string

/CREATE
/NODIAGNOSTICS
/NOLIST (Interactive)
/LIST (Batch)
/NOLOG
/NOPRINT

1.1.16 SAVE Command (ADU>)

Puts the last command you entered in the file you designate.

SAVE save-file-spec

1.1.17 SET DEFAULT Command (ADU>)

Assigns your default directory in the dictionary.

SET DEFAULT cdd-path-spec

1.1.18 SET LOG Command (ADU>)

Creates a log file of an interactive ADU session you enable with the SET LOG command. The SET NOLOG command disables logging.

SET LOG [log-file-spec]

SET NOLOG

1.1.19 SET VERIFY Command (ADU>)

Displays commands and source definitions as they are processed from a command file you execute with the @ (At sign) command. The SET NOVERIFY command disables the displaying of processed commands and source definitions.

SET VERIFY

SET NOVERIFY

1.1.20 SHOW DEFAULT Command (ADU>)

Displays your current default dictionary directory.

SHOW DEFAULT

1.1.21 SHOW LOG Command (ADU>)

Displays information about logging you enable with the SET LOG command. The SET NOLOG command disables logging.

SHOW LOG

1.1.22 SHOW VERSION Command (ADU>)

Displays the current software version number of ADU.

SHOW VERSION

1.1.23 SPAWN Command (ADU>)

Creates a subprocess of the current process and transfers job control to the subprocess.

SPAWN [command] [/qualifiers]

Command Qualifiers	Defaults
/INPUT=file-spec	/INPUT=SYSS\$INPUT
/[NO]LOGICAL_NAMES	/LOGICAL_NAMES
/OUTPUT=file-spec	/OUTPUT=SYS\$OUTPUT
/NOOUTPUT	
/PROCESS [=subprocess-name]	
/[NO]SYMBOLS	/SYMBOLS
/[NO]WAIT	/WAIT

ADU Commands and Clauses

1.2 %INCLUDE

1.2 %INCLUDE

Many definitions share common parts. For example, suppose you always include certain default characteristics in an application definition. Instead of rewriting the same part of a definition many times, you can use %INCLUDE to put the contents of a file in a source definition.

1.2.1 %INCLUDE

Includes the contents of a file in a source definition. If you do not specify a file type, ACMS supplies the .COM default.

```
%INCLUDE "file-spec"
```

1.3 Task Definition Clauses

This section lists the syntax for the ADU clauses and phrases you use to write task definitions. You use these clauses with the ADU CREATE, MODIFY, REPLACE, or EDIT commands.

A **task definition** is made up of clauses describing the attributes of a task and the work done when a user selects a task. **Task attribute clauses** can either define the implementation characteristics or the control attributes of a task.

Task attribute clauses describe general characteristics of a task, such as the workspaces used by task steps or a default server that handles processing work. You can override some characteristics by specifying the same clause with a different attribute in a step definition.

The work part of a task is defined either in a processing step or a block step made up of processing and exchange steps. You can define a single-step task or use the BLOCK WORK clause to define multiple-step tasks. ACMS lets you nest block steps so that a task can contain multiple blocks.

This section begins with overview syntax for tasks, block step phrases, exchange steps, processing steps, action clauses, and exception handler action clauses. The overview syntax is followed by syntax for individual task definition clauses and phrases.

1.3.1 Task Syntax

[DEFAULT REQUEST LIBRARY IS request-library-name;]

[DEFAULT FORM IS form-label-name;]

[DEFAULT SERVER IS server-name ;]

[[NO] DELAY ;
[NO] WAIT ;]

[LOCAL ;
GLOBAL ;]

[[NOT] CANCELABLE BY [[TERMINAL] USER
[TASK] SUBMITTER] ;]

[USE { WORKSPACE
WORKSPACES }
workspace-name
{ [WITH ACCESS { RETRIEVAL
UPDATE [[NO] LOCK] }] } [, ...] ;] ...

[{ WORKSPACE IS
WORKSPACES ARE }
record-path-name
{ WITH { NAME unique-name
TYPE { GROUP
TASK
USER }
ACCESS { RETRIEVAL
UPDATE [[NO] LOCK] } } } [, ...] ;] ...

ADU Commands and Clauses

1.3 Task Definition Clauses

```

[ TASK { ARGUMENT IS
        ARGUMENTS ARE }
  { workspace-name
    { [ WITH ACCESS { READ
                    WRITE
                    MODIFY } ] } } [ ... ] ;

```

```

{ BLOCK WORK [ WITH <block-phrase> ... ] IS
  [ <block-conditional-clause> ]
  { [ label: ] { { BLOCK WORK [ WITH <block-phrase> ] IS
                <block-step> }
                { EXCHANGE WORK IS
                  <exchange-clause> }
                { PROCESSING WORK
                  [ WITH <processing-phrase> ... ] IS
                  <processing-clause> ... } } } ...
  [ ACTION IS
    <action-clause> ... ]
  [ EXCEPTION HANDLER ACTION IS
    <action-clause> ... ]
  END BLOCK WORK ;
  [ ACTION IS
    <action-clause> ... ]
  [ EXCEPTION HANDLER ACTION IS
    <action-clause> ... ]
  { PROCESSING WORK [ WITH <processing-phrase> ... ] IS
    <processing-clause>
    [ ACTION IS
      <action-clause> ... ]
    [ EXCEPTION HANDLER ACTION IS
      <action-clause> ... ]
  }
}

```

1.3.2 Block Step Phrases Syntax

BLOCK WORK

$$\left[\begin{array}{l} \left\{ \begin{array}{l} \text{CANCEL ACTION IS } \langle \text{processing-clause} \rangle \\ \text{DISTRIBUTED TRANSACTION} \\ \left\{ \begin{array}{l} \text{NO TERMINAL USER I/O} \\ \text{REQUEST I/O} \\ \text{FORM I/O} \\ \text{STREAM I/O} \end{array} \right\} \\ \text{[NO] SERVER CONTEXT} \end{array} \right\} \text{ WITH } \left. \vphantom{\left\{ \begin{array}{l} \text{CANCEL ACTION IS } \langle \text{processing-clause} \rangle \\ \text{DISTRIBUTED TRANSACTION} \\ \left\{ \begin{array}{l} \text{NO TERMINAL USER I/O} \\ \text{REQUEST I/O} \\ \text{FORM I/O} \\ \text{STREAM I/O} \end{array} \right\} \\ \text{[NO] SERVER CONTEXT} \end{array} \right\}} \right\} \text{ IS } \end{array} \right]$$

[<block-conditional-clause>]

$$\left[\begin{array}{l} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{BLOCK WORK } \left[\text{WITH } \left[\begin{array}{l} \text{DISTRIBUTED TRANSACTION} \\ \text{[NO] SERVER CONTEXT} \end{array} \right] \right] \text{ IS } \\ \langle \text{block-step} \rangle \end{array} \right\} \\ \left\{ \begin{array}{l} \text{EXCHANGE WORK IS} \\ \langle \text{exchange-clause} \rangle \end{array} \right\} \\ \left\{ \begin{array}{l} \text{PROCESSING WORK} \\ \left[\text{WITH } \langle \text{processing-phrase} \rangle \dots \right] \text{ IS} \\ \langle \text{processing-clause} \rangle \dots \end{array} \right\} \end{array} \right\} \dots \\ \left[\begin{array}{l} \text{ACTION IS} \\ \langle \text{action-clause} \rangle \dots \end{array} \right] \\ \left[\begin{array}{l} \text{EXCEPTION HANDLER ACTION IS} \\ \langle \text{action-clause} \rangle \dots \end{array} \right] \end{array} \right]$$

END BLOCK WORK ;

[ACTION IS
 <action-clause> ...]

[EXCEPTION HANDLER ACTION IS
 <action-clause> ...]

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.3 Exchange Step Syntax

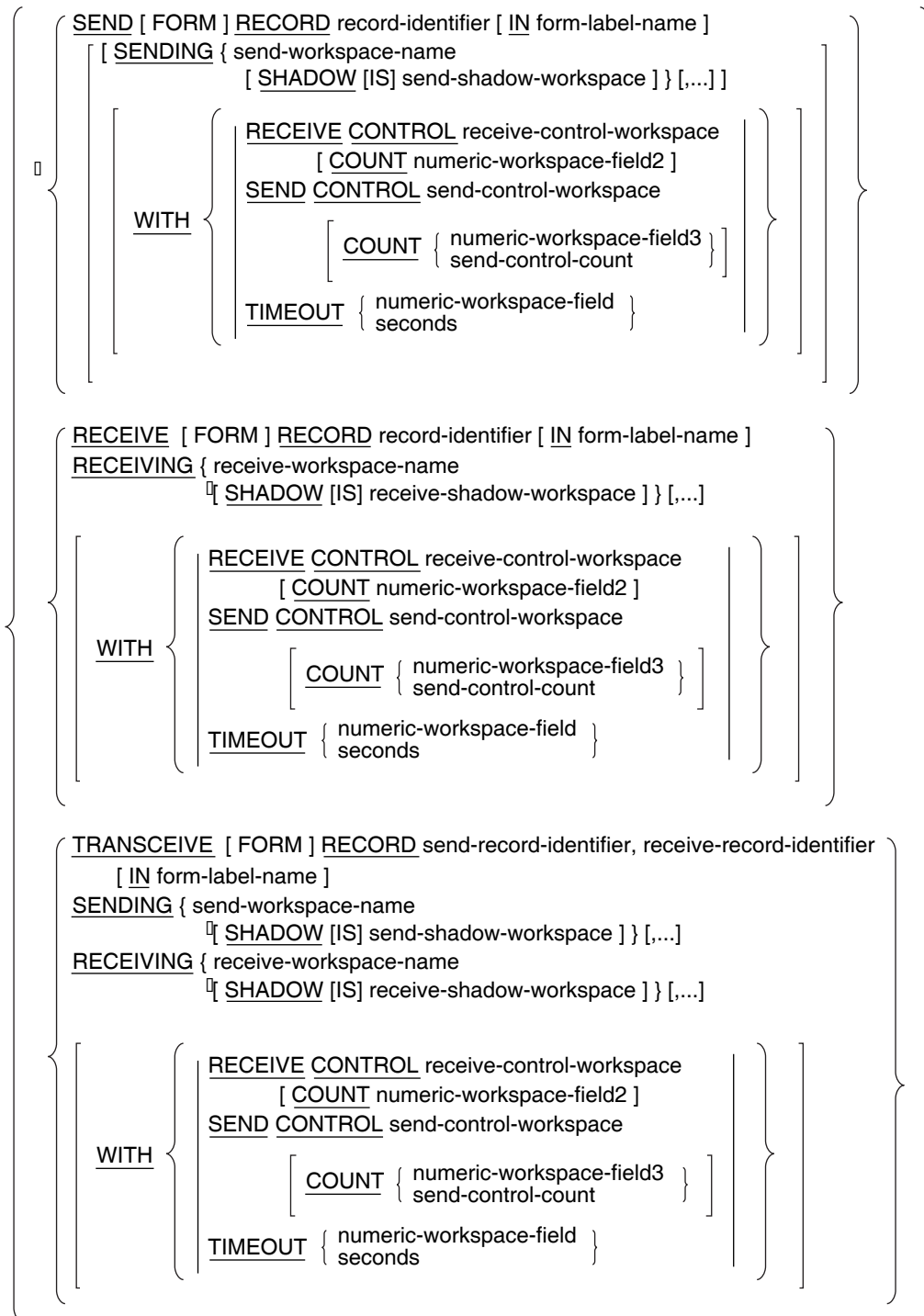
EXCHANGE WORK IS

```
{  
  {  
    CONTROL FIELD control-field  
    {  
      value : <exchange-clause> [...]  
      NOMATCH : <exchange-clause>  
    }  
  }  
  END CONTROL FIELD ;  
  
  {  
    IF (boolean-expression)  
      THEN <exchange-clause>  
      [ ELSE <exchange-clause> ]  
  }  
  END IF ;  
  
  {  
    SELECT FIRST TRUE OF  
    {  
      (boolean-expression) : <exchange-clause> [...]  
      NOMATCH : <exchange-clause>  
    }  
  }  
  END SELECT ;  
  
  {  
    WHILE (boolean-expression)  
    DO <exchange-clause>  
  }  
  END WHILE ;  
  
  { NO EXCHANGE ; }  
  
  {  
    READ read-workspace-name  
    [ WITH PROMPT { prompt-workspace-name } ] ;  
  }  
  
  {  
    REQUEST IS request-name [ IN request-library ]  
    [ USING workspace-name [ ,... ] ;  
  }  
  
  {  
    WRITE { workspace-name } ;  
  }  
}
```

TAY-0403-AI

ADU Commands and Clauses

1.3 Task Definition Clauses



TAY-0403A-AI

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.4 Processing Step Syntax

PROCESSING WORK

```

WITH
{
  {
    { DISTRIBUTED TRANSACTION }
    { NONPARTICIPATING SERVER }
  }
  {
    { [NO] TERMINAL USER I/O }
    { REQUEST I/O }
  }
} IS

{
  {
    { CONTROL FIELD control-field
      { value : <processing-clause> [... ] }
      { NOMATCH : <processing-clause> }
    }
    { END CONTROL FIELD ;
  }

  {
    { IF (boolean-expression)
      { THEN <processing-clause>
      { [ELSE <processing-clause> ]
    }
    { END IF ;
  }

  {
    { SELECT FIRST TRUE OF
      { (boolean-expression) : <processing-clause> [... ] }
      { NOMATCH : <processing-clause>
    }
    { END SELECT ;
  }

  {
    { WHILE (boolean-expression)
      { DO <processing-clause>
    }
    { END WHILE ;
  }

  {
    { CALL [PROCEDURE] entry-point-name [ IN server-name ]
      { [USING workspace-name [ ,... ] ] ;
  }

  { CALL TASK task-name [ USING workspace-name [ ,... ] ] ;

  {
    { { DATATRIEVE }
      { { DTR } }
    } COMMAND IS dtr-command-string [ IN server-name ] ;
  }

  { DCL COMMAND IS dcl-command-string [ IN server-name ] ;
  { IMAGE IS image-file-spec [ IN server-name ] ;
  { NO PROCESSING ;
}

```

1.3.5 Action Clauses Syntax

ACTION IS

```

CONTROL FIELD control-field {value : <action-clause>...}... END CONTROL FIELD ;
IF (boolean-expression) THEN <action-clause>... [ ELSE <action-clause> ] END IF;
SELECT FIRST TRUE OF {(boolean-expression) : <action-clause>...}... END SELECT ;

[ GET MESSAGE [ NUMBER { message-number
workspace-field
global-symbol } ] [INTO workspace-field];

MOVE { { signed-number
workspace-field
quoted-string
global-symbol } { INTO
TO } { (workspace-field [...])
workspace-field } } [...];

{ COMMIT TRANSACTION;
ROLLBACK TRANSACTION; }

{ NO SERVER CONTEXT ACTION ;
RELEASE SERVER CONTEXT [ IF ACTIVE SERVER CONTEXT ] ;
RETAIN SERVER CONTEXT [ IF ACTIVE SERVER CONTEXT ] ; }

[ CANCEL TASK [ RETURNING { message-number
numeric-workspace-field
global-symbol } ] ;

EXIT BLOCK ;

EXIT TASK [ RETURNING { message-number
numeric-workspace-field
global-symbol } ] ;

[ GO TO ] { STEP step-label-name
[ GOTO ] { { NEXT
PREVIOUS } { EXCHANGE
PROCESSING
STEP } } ;

REPEAT STEP ;

RAISE EXCEPTION [ { message-number
numeric-workspace-field
global-symbol } ] ;

```

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.6 BLOCK Clause (Block)

Describes the work done in a block step in terms of block, exchange, processing, and action clauses.

```
BLOCK WORK [ WITH <block-phrase> ... ] IS
```

```
[ <block-conditional-clause> ]
```

```
[ label: ]
{
  {
    { BLOCK WORK [ WITH <block-phrase> ... ] IS }
    <block-step>
  }
  {
    { EXCHANGE WORK IS }
    <exchange-clause>
  }
  {
    { PROCESSING WORK
      [ WITH <processing-phrase> ... ] IS }
    <processing-clause> ...
  }
  ...
  [ ACTION IS
    <action-clause> [,...] ]
  [ EXCEPTION HANDLER ACTION IS
    <action-clause> [,...] ]
}
```

```
END BLOCK WORK ;
```

```
[ ACTION IS
  <action-clause> [,...] ]
```

```
[ EXCEPTION HANDLER ACTION IS
  <action-clause> [,...] ]
```

1.3.7 CALL Clause (Processing)

Names a procedure in a procedure server to do the work for a processing step. Also names any workspaces used by that procedure.

```
CALL PROCEDURE entry-point-name [ IN server-name ] [ USING workspace-name [,... ] ];
```

1.3.8 CALL TASK Clause (Processing)

Names a task called by a processing step and any workspaces supplied to the called task.

```
CALL TASK task-name [ USING workspace-name [ ,... ] ] ;
```

1.3.9 CANCEL ACTION Phrase (Block)

Specifies processing ACMS does when a task is canceled.

```
CANCEL ACTION IS <processing-clause>
```


1.3.10 CANCEL TASK Clause (Action)

Stops the task in the action part of the current step by canceling the current task instance.

$$\underline{\text{CANCEL TASK}} \left[\underline{\text{RETURNING}} \left\{ \begin{array}{l} \text{message-number} \\ \text{numeric-workspace-field} \\ \text{global-symbol} \end{array} \right\} \right] ;$$

1.3.11 CANCELABLE Clause (Task)

Specifies whether or not a task can be canceled by a user or task submitter.

$$[\text{NOT}] \underline{\text{CANCELABLE}} \text{ BY } \left[\begin{array}{l} [\text{TERMINAL}] \text{ USER} \\ [\text{TASK}] \text{ SUBMITTER} \end{array} \right] ;$$

1.3.12 COMMIT TRANSACTION Clause (Action)

Signals the end of a distributed transaction and makes permanent any file or database operations performed within the distributed transaction.

COMMIT TRANSACTION;

1.3.13 CONTROL FIELD Clause (Action, Block, Exchange, Processing)

Performs a step or action based on a condition.

$$\underline{\text{CONTROL FIELD}} \text{ control-field} \\ \left\{ \begin{array}{l} \text{value : } <\text{clause}> [, \dots] \\ \text{NOMATCH : } <\text{clause}> [, \dots] \end{array} \right\} \\ \underline{\text{END CONTROL FIELD}} ;$$

1.3.14 DATATRIEVE COMMAND Clause (Processing)

Names a DATATRIEVE command to do work for a processing step.

$$\left\{ \begin{array}{l} \underline{\text{DATATRIEVE}} \\ \underline{\text{DTR}} \end{array} \right\} \underline{\text{COMMAND}} \text{ IS dtr-command-string [IN server-name] ;$$

1.3.15 DCL COMMAND Clause (Processing)

Names a DCL command to do work for a processing step.

DCL COMMAND IS dcl-command-string [IN server-name] ;

1.3.16 DEFAULT FORM Clause (Task)

Names a default form used by the SEND, RECEIVE, and TRANSCEIVE clauses in exchange steps of a task.

DEFAULT FORM IS form-label-name;

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.17 DEFAULT REQUEST LIBRARY Clause (Task)

Names a default request library used by REQUEST clauses in exchange steps of a task.

DEFAULT REQUEST LIBRARY IS request-library-name ;

1.3.18 DEFAULT SERVER Clause (Task)

Names a default server to handle processing and cancel actions for the step or steps in a task.

DEFAULT SERVER IS server-name ;

1.3.19 DELAY Clause (Task)

Controls whether or not ACMS pauses after a task finishes running before clearing the screen and displaying the ACMS menu.

[NO] DELAY ;

1.3.20 EXCEPTION HANDLER Clause (Block, Exchange, Processing)

Describes the actions to be taken to recover from one or more exceptions.

EXCEPTION HANDLER ACTION IS <action-clause> ... ;

1.3.21 EXCHANGE Clause (Task)

Describes the interaction between the application and the user.

EXCHANGE WORK IS

<exchange-clause>

[ACTION IS
 <action-clause> [,...]]

[EXCEPTION HANDLER ACTION IS
 <action-clause> ...]

1.3.22 EXIT BLOCK Clause (Action)

Transfers control of the task to the action part of the block step definition.

EXIT BLOCK ;

1.3.23 EXIT TASK Clause (Action)

Ends the current task.

$$\underline{\text{EXIT TASK}} \left[\underline{\text{RETURNING}} \left\{ \begin{array}{l} \text{message-number} \\ \text{numeric-workspace-field} \\ \text{global-symbol} \end{array} \right\} \right];$$

1.3.24 FORM I/O Phrase (Block)

Specifies that the exchange steps in a block step use HP DECforms to interface with the user.

FORM I/O

1.3.25 GET ERROR MESSAGE Clause (Action)

Uses the OpenVMS message facility to translate a message number into a message and move that message from a message file to a workspace field.

$$\underline{\text{GET ERROR MESSAGE}} \left[\underline{\text{NUMBER}} \left\{ \begin{array}{l} \text{message-number} \\ \text{numeric-workspace-field} \\ \text{global-symbol} \end{array} \right\} \right] \\ \left[\underline{\text{INTO}} \text{ workspace-string-field } \right];$$

1.3.26 GLOBAL Clause (Task)

Specifies that a task can be selected from a menu, called by an agent, or called by another task.

GLOBAL ;

1.3.27 GOTO STEP Clause (Action)

In the action part of a step definition, specifies which block, exchange, or processing step to execute next.

$$\left[\begin{array}{l} \underline{\text{GO TO}} \\ \underline{\text{GOTO}} \end{array} \right] \left\{ \begin{array}{l} \{ \underline{\text{STEP}} \text{ step-label-name } \} \\ \left\{ \begin{array}{l} \underline{\text{NEXT}} \\ \underline{\text{PREVIOUS}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{EXCHANGE}} \\ \underline{\text{PROCESSING}} \\ \underline{\text{STEP}} \end{array} \right\} \end{array} \right\};$$

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.28 IF THEN ELSE Clause (Action, Block, Exchange, Processing)

Takes action based on values you test with Boolean expressions. Use the IF THEN ELSE clause to start a block, exchange, or processing step (thereby creating a conditional block, exchange, or processing step), or to start an action clause (thereby creating a conditional action clause).

```
IF (boolean-expression)
  THEN <clause>
  [ ELSE <clause> ]
END IF;
```

1.3.29 IMAGE Clause (Processing)

Names an OpenVMS image to do work for a processing step.

```
IMAGE IS image-file-spec [ IN server-name ] ;
```

1.3.30 LOCAL Clause (Task)

Specifies that a task can be called by or chained to another task, but not selected from a menu or called by an agent.

```
LOCAL ;
```

1.3.31 MOVE Clause (Action)

Specifies that a number, the numeric value of a global symbol, workspace field, or quoted string is to move into another workspace field or fields.

```
MOVE { { signed-number
          global-symbol
          workspace-field
          quoted-string } { INTO
                          } { (workspace-field [...])
                              workspace-field } } [...];
```

1.3.32 NO EXCHANGE Clause (Exchange)

Specifies that an exchange step does not do any work.

```
NO EXCHANGE ;
```

1.3.33 NO PROCESSING Clause (Processing)

Specifies that the step does not do any processing work.

```
NO PROCESSING ;
```

1.3.34 NO SERVER CONTEXT ACTION Clause (Action)

Maintains the current state of any server context associated with the task.

```
NO SERVER CONTEXT ACTION ;
```

1.3.35 NO TERMINAL I/O Phrase (Block, Processing)

States that the block or processing step does no terminal I/O.

NO TERMINAL USER I/O

1.3.36 NONPARTICIPATING SERVER Phrase (Processing)

Excludes a processing step from participating in an existing distributed transaction.

NONPARTICIPATING SERVER

1.3.37 PROCESSING Clause (Task)

Describes work done in a single-step processing task.

PROCESSING WORK [WITH <processing-phrase> [,...] IS

<processing-clause>

[ACTION IS
 <action-clause> [,...]]

[EXCEPTION HANDLER ACTION IS]
 <action-clause> ...]

1.3.38 RAISE EXCEPTION Clause (Action)

Raises a step exception and passes control to the exception handler action part of the step.

RAISE EXCEPTION [{ message-number
 numeric-workspace-field
 global-symbol }] ;

1.3.39 READ Clause (Exchange)

If the block step uses STREAM I/O, the READ clause reads from an ACMS stream into a workspace. If the block step uses REQUEST I/O, the READ clause passes information from the exception line (line 24) on the terminal screen to a workspace.

READ read-workspace-name [WITH PROMPT { prompt-workspace-name
 literal-string }] ;

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.40 RECEIVE Clause (Exchange)

Transfers information from form data items to your task workspace.

```
RECEIVE [ FORM ] RECORD record-identifier [ IN form-label-name ]
RECEIVING { receive-workspace-name
           [ SHADOW [IS] receive-shadow-workspace ] } [...]
```

WITH {

```
    RECEIVE CONTROL receive-control-workspace
           [ COUNT numeric-workspace-field2 ]
    SEND CONTROL send-control-workspace
           [ COUNT { numeric-workspace-field3
                     send-control-count } ]
    TIMEOUT { numeric-workspace-field
              seconds }
```

}

1.3.41 RELEASE SERVER CONTEXT Clause (Action)

Releases the server process allocated for a task.

```
RELEASE SERVER CONTEXT [ IF ACTIVE SERVER CONTEXT ] ;
```

1.3.42 REPEAT STEP Clause (Action)

Repeats the current exchange, processing, or block step.

```
REPEAT STEP ;
```

1.3.43 REQUEST Clause (Exchange)

Names a TDMS request that does input and output for an exchange step.

```
REQUEST IS request-name [ IN request-library ] [ USING workspace-name [...] ] ;
```

1.3.44 REQUEST I/O Phrase (Block, Processing)

Specifies using TDMS to communicate with the user. Specify the REQUEST I/O phrase at the block or processing step level.

```
REQUEST I/O
```

1.3.45 RETAIN SERVER CONTEXT Clause (Action)

Retains the server context within the current server.

```
RETAIN SERVER CONTEXT [ IF ACTIVE SERVER CONTEXT ] ;
```

1.3.46 ROLLBACK TRANSACTION Clause (Action)

Marks the end of a distributed transaction and returns any files and databases within the transaction to the state they were in before the transaction started.

ROLLBACK TRANSACTION ;

1.3.47 SELECT FIRST Clause (Action, Block, Exchange, Processing)

Takes action based on values you test with Boolean expressions. Use a SELECT FIRST clause to start a block, exchange, or processing step (thereby creating a conditional block, exchange, or processing step), or to start an action clause (thereby creating a conditional action clause).

```
SELECT FIRST TRUE OF
{
  (boolean-expression) : <clause> [ ,... ]
  NOMATCH              : <clause> [ ,... ]
}
END SELECT ;
```

1.3.48 SEND Clause (Exchange)

Transfers information from your task workspace to form data items.

```
SEND [ FORM ] RECORD record-identifier [ IN form-label-name ]
{
  [ SENDING { send-workspace-name
    [ SHADOW [IS] send-shadow-workspace ] } [,... ]
  ]
  [
    [
      [
        [
          RECEIVE CONTROL receive-control-workspace
          [ COUNT numeric-workspace-field2 ]
          SEND CONTROL send-control-workspace
        ]
        [
          [
            [
              COUNT { numeric-workspace-field3
                send-control-count
              }
            ]
          ]
        ]
      ]
      [
        [
          TIMEOUT { numeric-workspace-field
            seconds
          }
        ]
      ]
    ]
  ]
}
}
```

1.3.49 SERVER CONTEXT Phrase (Block)

Specifies whether or not server context is retained by default between steps in a block step.

[NO] SERVER CONTEXT

1.3.50 STREAM I/O Phrase (Block)

Specifies that the exchange steps in a block step use ACMS streams to communicate with the user or other task submitter.

STREAM I/O

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.51 TASK ARGUMENTS Phrase (Task)

Identifies the names and the order of the task workspace arguments that can be supplied to a called task by an agent or by another task.

$$\text{TASK } \left\{ \begin{array}{l} \text{ARGUMENT IS} \\ \text{ARGUMENTS ARE} \end{array} \right\}$$

$$\left\{ \text{workspace-name } \left[\text{WITH ACCESS } \left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \\ \text{MODIFY} \end{array} \right\} \right] \right\} [\dots] ;$$

1.3.52 TERMINAL I/O Phrase (Processing)

Specifies that a processing step communicates directly with the terminal by means of programming statements, OpenVMS services, or TDMS requests.

TERMINAL I/O

1.3.53 TRANSACTION Phrase (Block, Processing)

Identifies the block or processing step as a transaction; all work within the step either completes successfully or is rolled back.

DISTRIBUTED TRANSACTION

1.3.54 TRANSCEIVE Clause (Exchange)

Combines the SEND and RECEIVE operations. First, HP DECforms sends data from your task workspace to form data items. Then it moves data from the form to your task workspace.

$$\left\{ \begin{array}{l} \text{TRANSCEIVE } [\text{FORM}] \text{ RECORD } \text{send-record-identifier, receive-record-identifier} \\ \quad [\text{IN form-label-name}] \\ \text{SENDING } \{ \text{send-workspace-name} \\ \quad [\text{SHADOW } [\text{IS}] \text{ send-shadow-workspace}] \} [\dots] \\ \text{RECEIVING } \{ \text{receive-workspace-name} \\ \quad [\text{SHADOW } [\text{IS}] \text{ receive-shadow-workspace}] \} [\dots] \\ \\ \left[\begin{array}{l} \text{WITH } \left\{ \begin{array}{l} \text{RECEIVE CONTROL } \text{receive-control-workspace} \\ \quad [\text{COUNT } \text{numeric-workspace-field2}] \\ \text{SEND CONTROL } \text{send-control-workspace} \\ \\ \left[\text{COUNT } \left\{ \begin{array}{l} \text{numeric-workspace-field3} \\ \text{send-control-count} \end{array} \right\} \right] \\ \\ \text{TIMEOUT } \left\{ \begin{array}{l} \text{numeric-workspace-field} \\ \text{seconds} \end{array} \right\} \end{array} \right. \end{array} \right. \end{array} \right\}$$

1.3.55 USE WORKSPACE Clause (Task)

Names one or more workspaces, declared in the task group, that a task needs to access.

$$\underline{\text{USE}} \left\{ \begin{array}{l} \underline{\text{WORKSPACE}} \\ \underline{\text{WORKSPACES}} \end{array} \right\}$$

$$\left\{ \text{workspace-name} \left[\underline{\text{WITH ACCESS}} \left\{ \begin{array}{l} \underline{\text{RETRIEVAL}} \\ \underline{\text{UPDATE}} \left[\underline{\text{NO}} \right] \underline{\text{LOCK}} \right] \right\} \right] \right\} \left[\dots \right];$$

1.3.56 WAIT Clause (Task)

Controls whether or not ACMS displays a message prompting users to press Return. Pressing Return clears the terminal screen and displays the previous ACMS menu.

[NO] WAIT ;

1.3.57 WHILE DO Clause (Block, Exchange, Processing)

Performs block, exchange, or processing work as long as a specified Boolean expression evaluates to true.

WHILE (boolean-expression)
DO <clause>
END WHILE;

1.3.58 WORKSPACES Clause (Task)

Names one or more workspaces used by steps in a task.

$$\left\{ \begin{array}{l} \underline{\text{WORKSPACE IS}} \\ \underline{\text{WORKSPACES ARE}} \end{array} \right\}$$

$$\left\{ \begin{array}{l} \text{record-path-name} \\ \left[\underline{\text{WITH}} \left\{ \begin{array}{l} \underline{\text{NAME}} \text{ unique-name} \\ \underline{\text{TYPE}} \left\{ \begin{array}{l} \underline{\text{GROUP}} \\ \underline{\text{TASK}} \\ \underline{\text{USER}} \end{array} \right\} \\ \underline{\text{ACCESS}} \left\{ \begin{array}{l} \underline{\text{RETRIEVAL}} \\ \underline{\text{UPDATE}} \left[\underline{\text{NO}} \right] \underline{\text{LOCK}} \right] \end{array} \right\} \end{array} \right\} \right] \end{array} \right\} \left[\dots \right];$$

ADU Commands and Clauses

1.3 Task Definition Clauses

1.3.59 WRITE Clause (Exchange)

If the block step uses STREAM I/O, the WRITE clause writes the contents of a workspace field to a stream. If the block step uses REQUEST I/O, the WRITE clause passes a literal string or the contents of a workspace to the exception line (line 24) on the terminal screen.

```
WRITE { workspace-name } ;  
        { literal-string }
```

1.4 Task Group Definition Clauses

A task group is a set of tasks that share resources and are built into a single database file. This section lists the syntax for the ADU clauses and subclauses used to define task groups.

Use task group clauses to define:

- Characteristics applying to all tasks in the group
- Servers that handle the processing for tasks in the group

Also use task group clauses to name the tasks belonging to the group and to define some tasks directly in the task group definition.

You can use two kinds of subclauses with task group clauses: **processing subclauses** and **server subclauses**. If a task consists of a single processing step, you can include the definition for the task directly in the task group definition. The task group clauses used to define a task directly in a task group definition are called processing subclauses.

When you define a server in a task group definition, use server subclauses to describe characteristics for that server.

You can define a task directly in a task group definition if that task:

- Consists of a single unconditional processing step
- Defines no step actions or exception handler actions
- Defines no default server or default form file
- Uses no workspaces other than the ACMS system workspaces

If a task definition does not follow these rules, name it in the task group definition. Define it separately, using task and block clauses.

This section begins with overview syntax for task groups, processing subclauses, and server subclauses. The overview syntax is followed by syntax for individual task group definition clauses and subclauses.

ADU Commands and Clauses

1.4 Task Group Definition Clauses

1.4.1 Task Group Syntax

```

[ DEFAULT TASK GROUP FILE IS task-group-database-file ;]

[ MESSAGE [ FILE IS
  FILES ARE ] message-file-spec [...]; ] ...

[ REQUEST { LIBRARY IS
  LIBRARIES ARE }
  { request-library-file-spec [ WITH NAME library-name ] } [...]; ] ...

[ { FORM IS
  FORMS ARE }
  form-name IN form-file-spec WITH NAME form-label-name [...]; ] ...

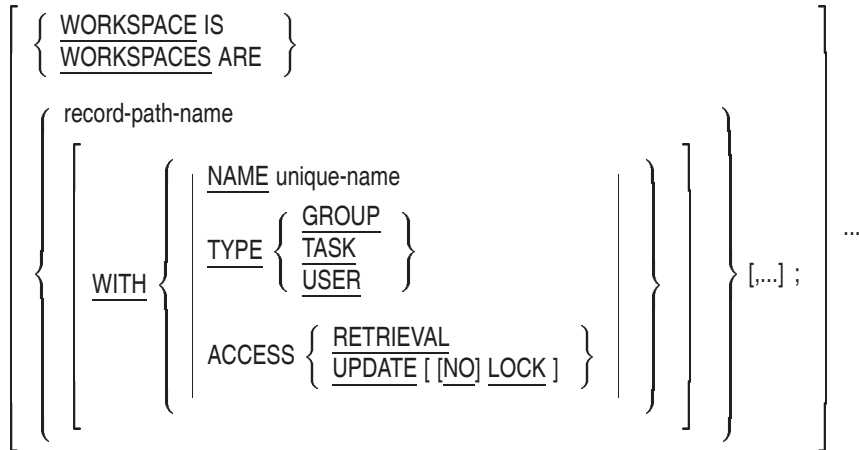
{ { SERVER IS
  SERVERS ARE } { server-name : <server-subclause> ... } ... } ...
END [ SERVER
  SERVERS ]; ]

{ { TASK IS
  TASKS ARE }
  task-name:
  { { [ [NO] DELAY;
    [ [NO] WAIT; ]
    [ LOCAL;
    [ GLOBAL; ]
    [ [NOT] CANCELABLE BY [ [TERMINAL] USER
    [ [TASK] SUBMITTER ] ];
    PROCESSING IS <processing-subclause>
    TASK DEFINITION IS task-path;
  } } } ... } ...
  END [ TASK
  TASKS ]; ]

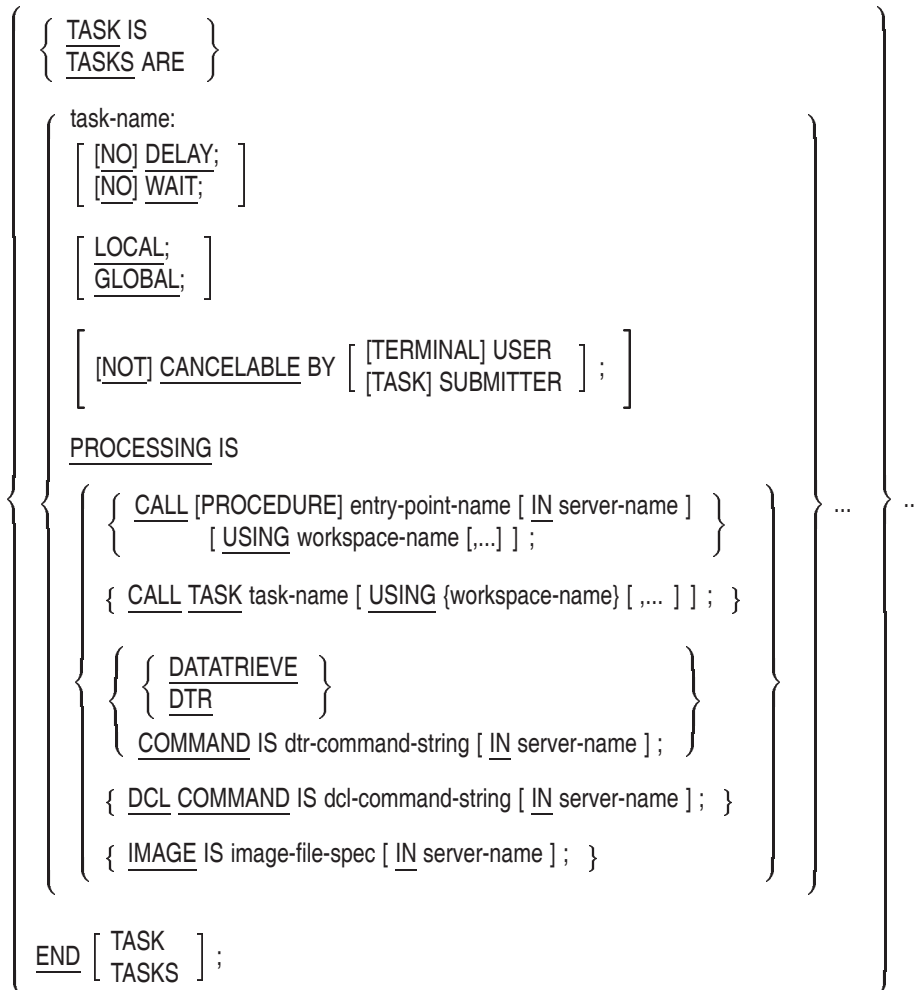
```

ADU Commands and Clauses

1.4 Task Group Definition Clauses



1.4.2 Processing Subclauses Syntax



1.4.3 Server Subclauses Syntax

```

{ SERVER IS
  SERVERS ARE }

server-name :
{
  { DCL PROCESS ; }
  { PROCEDURE SERVER IMAGE IS procedure-image-file-spec ;
    { ALWAYS EXECUTE TERMINATION PROCEDURE ON RUNDOWN ;
      CANCEL PROCEDURE IS cancel-entry-name ;
      DEFAULT OBJECT FILE IS object-file-spec ;
      { { INITIALIZATION } PROCEDURE
        { INITIAL }
          IS initial-procedure-entry-name ; }
      { { PROCEDURE IS
        { PROCEDURES ARE } } ...
        { entry-name } [...] ; }
      [ RUNDOWN ON CANCEL [ IF INTERRUPTED ] ;
        NO RUNDOWN ON CANCEL ; ]
      { { TERMINATION } PROCEDURE
        { TERMINAL }
          IS terminal-procedure-entry-name ; }
      [ NO ] DCL AVAILABLE ; }
  }
  { [ USERNAME IS USERNAME OF TERMINAL USER ; ]
    [ NOT ] REUSABLE ;
    [ [ DYNAMIC USERNAME ; ]
      [ FIXED USERNAME ; ] ]
  }
}

END [ SERVER
  SERVERS ] ;

```

ADU Commands and Clauses

1.4 Task Group Definition Clauses

1.4.4 ALWAYS EXECUTE TERMINATION PROCEDURE Subclause (Server)

Specifies that ACMS should always process the server's termination procedure when the server process is run down.

ALWAYS EXECUTE TERMINATION PROCEDURE ON RUNDOWN;

1.4.5 CALL Subclause (Processing)

Names a procedure in a procedure server to do the work for a processing step.

CALL PROCEDURE entry-point-name [IN server-name] [USING workspace-name [...]];

1.4.6 CANCEL PROCEDURE Subclause (Server)

Names a procedure that runs when a task instance is canceled while that task is processing in a server or is maintaining server context in the server.

CANCEL PROCEDURE IS cancel-entry-name ;

1.4.7 DATATRIEVE COMMAND Subclause (Processing)

Runs a DATATRIEVE command to do work for a processing step.

{ DATATRIEVE } COMMAND IS dtr-command-string [IN server-name] ;
DTR

1.4.8 DCL AVAILABLE Subclause (Server)

Allows you to specify the loading of the DCL command line interpreter (CLI) into a procedure server process.

[NO] DCL AVAILABLE ;

1.4.9 DCL COMMAND Subclause (Processing)

Uses a DCL command to process a task.

DCL COMMAND [IS] dcl-command-string [IN server-name] ;

1.4.10 DCL PROCESS Subclause (Server)

Indicates that a server processes tasks that use DCL commands or command procedures, DATATRIEVE commands or procedures, or OpenVMS images.

DCL PROCESS ;

1.4.11 DEFAULT OBJECT FILE Subclause (Server)

Specifies a file name for the object module produced for a server when you build the task group containing that server.

DEFAULT OBJECT FILE IS object-file-spec ;

1.4.12 DEFAULT TASK GROUP FILE Clause (Task Group)

Names the default file specification of the task group database.

DEFAULT TASK GROUP FILE IS task-group-database-file ;

1.4.13 DYNAMIC USERNAME Subclause (Server)

Specifies that the user name, UIC, and default directory of a server change to match those of the user each time the server process is used.

DYNAMIC USERNAME ;

1.4.14 FIXED USERNAME Subclause (Server)

Specifies that the user name, UIC, and default directory of the server are those associated with the user name the server starts under.

FIXED USERNAME ;

1.4.15 FORMS Clause (Task Group)

Names the forms the task group uses.

{ FORM IS }
{ FORMS ARE }
form_name IN form-file-spec WITH NAME form-label-name [...] ;

1.4.16 IMAGE Subclause (Processing)

Names the OpenVMS image that ACMS runs when users select an image task.

IMAGE IS image-file-spec [IN server-name];

1.4.17 INITIALIZATION PROCEDURE Subclause (Server)

Names a procedure that runs when a procedure server image is started. An initialization procedure performs such activities as opening files used by the procedures handled by a server.

{ INITIALIZATION }
{ INITIAL } PROCEDURE IS initial-procedure-entry-name ;

ADU Commands and Clauses

1.4 Task Group Definition Clauses

1.4.18 MESSAGE FILES Clause (Task Group)

Names the message files used by the GET ERROR MESSAGE clause in the definitions of tasks in a task group.

MESSAGE [FILE IS
FILES ARE] message-file-spec [...] ;

1.4.19 PROCEDURE SERVER IMAGE Subclause (Server)

Identifies a server as a procedure server and names the procedure server image that does processing work for one or more tasks.

PROCEDURE SERVER IMAGE IS procedure-image-file-spec;

1.4.20 PROCEDURES Subclause (Server)

Names the step procedures that can run in a procedure server.

{ PROCEDURE IS
PROCEDURES ARE } entry-name [...] ;

1.4.21 REQUEST LIBRARIES Clause (Task Group)

Names the request libraries the task group uses.

REQUEST { LIBRARY IS
LIBRARIES ARE }
request-library-file-spec [WITH NAME library-name] [...] ;

1.4.22 REUSABLE Subclause (Server)

Identifies a server process as able or unable to process more than one processing step for more than one task without being restarted. Server processes that are not reusable must be started each time they are needed.

[NOT] REUSABLE ;

1.4.23 RUNDOWN ON CANCEL Subclause (Server)

Causes a procedure server to exit when a task cancel occurs while the task is keeping context in that server. When the server exits, ACMS releases server context. If you specify RUNDOWN ON CANCEL IF INTERRUPTED, ACMS runs down the server process only if ACMS interrupts the execution of a step procedure due to an exception.

[RUNDOWN ON CANCEL [IF INTERRUPTED] ;
NO RUNDOWN ON CANCEL ;]

1.4.24 **SERVERS Clause (Task Group)**

Defines the servers that handle the processing work for the tasks in a task group.

```

{ SERVER IS
  SERVERS ARE }
    { server-name : <server-subclause> [... ] } ...
END [ SERVER
        SERVERS ] ;

```

1.4.25 **TASKS Clause (Task Group)**

Identifies the tasks belonging to the task group you define.

```

{ TASK IS
  TASKS ARE }
    { task-name:
      {
        { [ [NO] DELAY;
          [NO] WAIT; ]
          { [ LOCAL;
            GLOBAL; ]
            { [ [NOT] CANCELABLE BY [ TERMINAL USER
              TASK SUBMITTER ] ;
              { PROCESSING IS <processing-subclause>
              { TASK DEFINITION IS task-path;
            }
          }
        }
      }
    } ...
END [ TASK
        TASKS ] ;

```

1.4.26 **TERMINATION PROCEDURE Subclause (Server)**

Names a procedure that runs when a procedure server image is stopped.

```

{ TERMINATION
  TERMINAL } PROCEDURE IS terminal-procedure-entry-name ;

```

1.4.27 **USERNAME Subclause (Server)**

Indicates that the server process runs under the OpenVMS user name of the user, and has the same UIC and default directory as that user.

```

USERNAME IS USERNAME OF TERMINAL USER ;

```

ADU Commands and Clauses

1.4 Task Group Definition Clauses

1.4.28 WORKSPACES Clause (Task Group)

Declares one or more workspaces used by the tasks in a task group.

```

{
  WORKSPACE IS
  WORKSPACES ARE
}

{
  record-path-name
  [
    WITH
    {
      NAME unique-name
      TYPE {
        GROUP
        TASK
        USER
      }
      ACCESS {
        RETRIEVAL
        UPDATE [ [NO] LOCK ]
      }
    }
  ]
} [...] ;

```

1.5 Application Definition Clauses

An application definition consists of a set of clauses that define control attributes for tasks, servers, and the application execution controller (EXC) that manages the server processes in which tasks run. This section lists the syntax for the ADU clauses and subclauses you use to define applications.

Two application definition clauses are required. The **TASK GROUPS** clause names the task group or groups that define the tasks of an application. The **APPLICATION USERNAME** clause defines the user name under which the application execution controller runs. The other clauses in the application definition are optional.

When ADU begins processing an application definition, it assigns default values to all characteristics of tasks and servers. You can change these default values by assigning different task characteristics to the tasks of an application with the **TASK ATTRIBUTES** or **TASK DEFAULTS** clause and by assigning different server characteristics to the servers of an application with the **SERVER ATTRIBUTES** or **SERVER DEFAULTS** clause.

This section begins with overview syntax for application definitions, server attributes clauses, server defaults clauses, task attributes clauses, and task defaults clauses. The overview syntax is followed by syntax for individual application definition clauses and subclauses.

1.5.1 Application Definition Syntax

$$\left[\text{APPLICATION } \underline{\text{DEFAULT DIRECTORY}} \text{ IS } \left\{ \begin{array}{l} \text{default-directory} \\ \underline{\text{USERNAME DEFAULT DIRECTORY}} \end{array} \right\} ; \right]$$

$$\left[\text{APPLICATION NAME } \left\{ \begin{array}{l} \underline{\text{TABLE IS}} \\ \underline{\text{TABLES ARE}} \end{array} \right\} \left\{ \begin{array}{l} \text{logical-name-table} \\ \text{quoted-string} \end{array} \right\} [\dots] ; \dots \right]$$

$$\left[\text{APPLICATION } \left\{ \begin{array}{l} \underline{\text{LOGICAL}} \\ \underline{\text{LOGICALS}} \end{array} \right\} \left[\begin{array}{l} \text{NAME IS} \\ \text{NAMES ARE} \end{array} \right] \right. \\ \left. \left\{ \left\{ \begin{array}{l} \text{logical-name} \\ \text{logical-string} \end{array} \right\} = \left\{ \begin{array}{l} \text{equivalence-name} \\ \text{equivalence-string} \end{array} \right\} \right\} [\dots] ; \dots \right]$$

{ APPLICATION USERNAME IS user-name ; }

[[NO] AUDIT;]

[DEFAULT APPLICATION FILE IS application-database-file ;]

$$\left[\left\{ \begin{array}{l} \underline{\text{MAXIMUM}} \\ \underline{\text{MAX}} \end{array} \right\} \text{SERVER } \left\{ \begin{array}{l} \underline{\text{PROCESS}} \\ \underline{\text{PROCESSES}} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{high-number} \\ \underline{\text{UNLIMITED}} \end{array} \right\} ; \right]$$

$$\left[\left\{ \begin{array}{l} \underline{\text{MAXIMUM}} \\ \underline{\text{MAX}} \end{array} \right\} \text{TASK } \left\{ \begin{array}{l} \underline{\text{INSTANCE}} \\ \underline{\text{INSTANCES}} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{high-number} \\ \underline{\text{UNLIMITED}} \end{array} \right\} ; \right]$$

$$\left[\underline{\text{SERVER CONTROL}} \left\{ \begin{array}{l} \underline{\text{ATTRIBUTE IS}} \\ \underline{\text{ATTRIBUTES ARE}} \end{array} \right\} \right. \\ \left. \left\{ \begin{array}{l} \text{server-given-name:} \\ \underline{\text{[SERVER group-server-name] [IN task-group-name]}} ; \dots \\ \text{[<server-subclause> ...]} \end{array} \right\} \dots \right] \\ \left[\underline{\text{END SERVER CONTROL}} \left[\begin{array}{l} \underline{\text{ATTRIBUTE}} \\ \underline{\text{ATTRIBUTES}} \end{array} \right] ; \right]$$

ADU Commands and Clauses

1.5 Application Definition Clauses

$$\left[\begin{array}{l} \text{SERVER } \left\{ \begin{array}{l} \text{DEFAULT IS} \\ \text{DEFAULTS ARE} \end{array} \right\} \text{ <server-subclause> ... } \\ \text{END SERVER } \left[\begin{array}{l} \text{DEFAULT} \\ \text{DEFAULTS} \end{array} \right]; \end{array} \right] \dots$$

[SERVER MONITORING INTERVAL IS seconds ;]

$$\left[\begin{array}{l} \text{TASK CONTROL } \left\{ \begin{array}{l} \text{ATTRIBUTE IS} \\ \text{ATTRIBUTES ARE} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{task-given-name:} \\ \text{[TASK group-task-name] [IN task-group-name];} \\ \text{[<task-subclause> ...];} \end{array} \right\} \dots \end{array} \right] \dots$$

$$\text{END TASK CONTROL } \left[\begin{array}{l} \text{ATTRIBUTE} \\ \text{ATTRIBUTES} \end{array} \right];$$

$$\left[\begin{array}{l} \text{TASK } \left\{ \begin{array}{l} \text{DEFAULT IS} \\ \text{DEFAULTS ARE} \end{array} \right\} \text{ <task-subclause> ... } \\ \text{END TASK } \left[\begin{array}{l} \text{DEFAULT} \\ \text{DEFAULTS} \end{array} \right]; \end{array} \right] \dots$$

$$\left\{ \begin{array}{l} \text{TASK } \left\{ \begin{array}{l} \text{GROUP IS} \\ \text{GROUPS ARE} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{task-group-given-name:} \\ \text{TASK GROUP FILE IS task-group-file ;} \end{array} \right\} \dots \end{array} \right\} \dots$$

$$\text{END TASK } \left[\begin{array}{l} \text{GROUP} \\ \text{GROUPS} \end{array} \right];$$

1.5.2 SERVER ATTRIBUTES Clause Syntax

SERVER CONTROL { ATTRIBUTE IS
ATTRIBUTES ARE }

server-name:

{

[NO] AUDIT;

[NO] { PROTECTED
PROTECT } { WORKSPACE
WORKSPACES } ;

[CREATION DELAY IS seconds;
CREATION INTERVAL IS seconds;]

DEFAULT DIRECTORY IS { default-directory
USERNAME DEFAULT DIRECTORY } ;

[DELETION DELAY IS seconds;
DELETION INTERVAL IS seconds;]

[[DYNAMIC USERNAME ;]
[FIXED USERNAME ;]]

[NAME] { TABLE IS
TABLES ARE } { logical-name-table
quoted-string } [...];

{ LOGICAL
LOGICALS } [NAME IS
NAMES ARE]

{ logical-name } = { equivalence-name
logical-string } = { equivalence-string } [...];

{ MAXIMUM
MAX } SERVER { PROCESS
PROCESSES } IS { high-number
UNLIMITED } ;

{ MINIMUM
MIN } SERVER { PROCESS
PROCESSES } IS low-number ;

[NO] [SERVER] PROCESS { DUMP
DUMPS } ;

USERNAME IS { username
USERNAME OF { TERMINAL USER
APPLICATION } } ;

END SERVER CONTROL [ATTRIBUTE
ATTRIBUTES] ;

}

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.3 SERVER DEFAULTS Clause Syntax

$$\begin{array}{l}
 \text{SERVER} \left\{ \begin{array}{l} \text{DEFAULT IS} \\ \text{DEFAULTS ARE} \end{array} \right\} \\
 [[\text{NO}] \text{AUDIT};] \\
 \left[[\text{NO}] \left\{ \begin{array}{l} \text{PROTECTED} \\ \text{PROTECT} \end{array} \right\} \left\{ \begin{array}{l} \text{WORKSPACE} \\ \text{WORKSPACES} \end{array} \right\}; \right] \\
 \left[\begin{array}{l} \text{CREATION DELAY IS seconds;} \\ \text{CREATION INTERVAL IS seconds;} \end{array} \right] \\
 \left[\text{DEFAULT DIRECTORY IS} \left\{ \begin{array}{l} \text{default-directory} \\ \text{USERNAME DEFAULT DIRECTORY} \end{array} \right\}; \right] \\
 \left[\begin{array}{l} \text{DELETION DELAY IS seconds;} \\ \text{DELETION INTERVAL IS seconds;} \end{array} \right] \\
 [[\text{DYNAMIC USERNAME};] \\
 [[\text{FIXED USERNAME};]] \\
 \left[[\text{NAME}] \left\{ \begin{array}{l} \text{TABLE IS} \\ \text{TABLES ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{logical-name-table} \\ \text{quoted-string} \end{array} \right\} [,\dots]; \right] \dots \\
 \left[\left\{ \begin{array}{l} \text{LOGICAL} \\ \text{LOGICALS} \end{array} \right\} [\text{NAME IS} \\
 \text{NAMES ARE}] \right. \\
 \left. \left\{ \left\{ \begin{array}{l} \text{logical-name} \\ \text{logical-string} \end{array} \right\} = \left\{ \begin{array}{l} \text{equivalence-name} \\ \text{equivalence-string} \end{array} \right\} \right\} [,\dots]; \right] \dots
 \end{array}$$

ADU Commands and Clauses

1.5 Application Definition Clauses

[{ MAXIMUM } SERVER { PROCESS PROCESSES } IS { high-number } ;]
[{ MAX } SERVER { PROCESS PROCESSES } IS { UNLIMITED } ;]

[{ MINIMUM } SERVER { PROCESS PROCESSES } IS low-number ;]
[{ MIN } SERVER { PROCESS PROCESSES } IS low-number ;]

[[NO] [SERVER] PROCESS { DUMP DUMPS } ;]
[[NO] [SERVER] PROCESS { DUMP DUMPS } ;]

[USERNAME IS { username } USERNAME OF { TERMINAL USER APPLICATION } ;]
[USERNAME IS { username } USERNAME OF { TERMINAL USER APPLICATION } ;]

END SERVER [DEFAULT DEFAULTS] ;

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.4 TASK ATTRIBUTES Clause Syntax

```

TASK CONTROL {
  ATTRIBUTE IS
  ATTRIBUTES ARE
}

task-name: [TASK group-task-name] [IN task-group-name];

{
  {
    ACCESS CONTROL LIST IS
    ( { IDENTIFIER } = acl-identifier [+...],
      { ID }
    )
    ACCESS = { EXECUTE }
              { NONE } ) [...];
  } ...
  [ NO ] AUDIT ;
  [ NO ] TRANSACTION TIMEOUT IS seconds ;
  [ NOT ] CANCELABLE BY [ [ TERMINAL ] USER
                        [ TASK ] SUBMITTER ] ;
  [ [ NO ] DELAY ;
    [ NO ] WAIT ; ]
  [ LOCAL ;
    GLOBAL ; ]
  [ { DISABLE } ;
    { DISABLED } ; ]
  [ { ENABLE } ;
    { ENABLED } ; ]
} ...

END TASK CONTROL [ ATTRIBUTE
                  ATTRIBUTES ] ;

```


1.5.5 TASK DEFAULTS Clause Syntax

```

TASK { DEFAULT IS
      DEFAULTS ARE }

{
  { ACCESS CONTROL LIST IS
    ( { IDENTIFIER
      ID } = acl-identifier [+...],
      ACCESS = { EXECUTE
                 NONE } ) [...]} ...
  [ NO ] AUDIT ;
  [ NO ] TRANSACTION TIMEOUT IS seconds;
  [ NOT ] CANCELABLE BY [ [ TERMINAL ] USER
                          [ TASK ] SUBMITTER ] ;

  [ [ NO ] DELAY ;
    [ NO ] WAIT ; ]

  [ LOCAL ;
    GLOBAL ; ]

  [ { DISABLE
      DISABLED } ;
    { ENABLE
      ENABLED } ; ]
}

END TASK [ DEFAULT
             DEFAULTS ] ;

```

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.6 ACCESS Subclause (Task)

Defines who can and cannot select a task.

ACCESS CONTROL LIST IS
({ IDENTIFIER } = acl-identifier [+...], ACCESS = { EXECUTE }) [...]
 { ID }

1.5.7 APPLICATION DEFAULT DIRECTORY Clause (Application)

Assigns a default device and directory for the process in which an application execution controller (EXC) runs.

APPLICATION DEFAULT DIRECTORY IS { default-directory
 USERNAME DEFAULT DIRECTORY } ;

1.5.8 APPLICATION LOGICALS Clause (Application)

Defines one or more process logical names for the process in which an application execution controller (EXC) runs.

APPLICATION { LOGICAL } [NAME IS
 LOGICALS] [NAMES ARE]

 { { logical-name } = { equivalence-name } }
 { { logical-string } = { equivalence-string } } } [...];

1.5.9 APPLICATION NAME TABLES Clause (Application)

Specifies one or more logical name tables the application execution controller (EXC) can use.

APPLICATION NAME { TABLE IS } { logical-name-table } [...];
 TABLES ARE } { quoted-string }

1.5.10 APPLICATION USERNAME Clause (Application)

Assigns an OpenVMS user name under which the application execution controller (EXC) runs.

APPLICATION USERNAME IS user-name ;

1.5.11 AUDIT Clause (Application, Server, Task)

Determines whether or not application, server, and/or task events are written to the ACMS Audit Trail Log.

[NO] AUDIT ;

1.5.12 CANCELABLE Subclause (Task)

Specifies whether or not a task can be canceled by a user or task submitter while the task is executing.

[NOT] CANCELABLE BY [[TERMINAL] USER
[TASK] SUBMITTER] ;

1.5.13 CREATION DELAY Subclause (Server)

Controls how long ACMS waits before beginning to create new server processes when tasks are waiting for a server process.

CREATION DELAY IS seconds ;

1.5.14 CREATION INTERVAL Subclause (Server)

Controls the intervals at which ACMS creates new server processes.

CREATION INTERVAL IS seconds ;

1.5.15 DEFAULT APPLICATION FILE Clause (Application)

Defines the application database file (.ADB) that ACMS uses when you do not name an application database file with the BUILD command.

DEFAULT APPLICATION FILE IS application-database-file;

1.5.16 DEFAULT DIRECTORY Subclause (Server)

Assigns a default device and directory for each of the server processes that are associated with a server.

DEFAULT DIRECTORY IS { default-directory
USERNAME DEFAULT DIRECTORY } ;

1.5.17 DELAY Subclause (Task)

Controls whether or not ACMS waits 3 seconds after a task finishes running before clearing the screen and displaying the ACMS menu.

[NO] DELAY ;

1.5.18 DELETION DELAY Subclause (Server)

Controls how long ACMS waits before deleting inactive server processes.

DELETION DELAY IS seconds ;

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.19 DELETION INTERVAL Subclause (Server)

Controls the intervals at which ACMS deletes inactive server processes.

DELETION INTERVAL IS seconds ;

1.5.20 DISABLE Subclause (Task)

Specifies that a task is not available for selection by task submitters.

{ DISABLE
DISABLED } ;

1.5.21 DYNAMIC USERNAME Subclause (Server)

Specifies that the user name, UIC, and default directory of a server change to match that of the user each time the server process is allocated for a task.

DYNAMIC USERNAME ;

1.5.22 ENABLE Subclause (Task)

Specifies that a task is available for selection by task submitters.

{ ENABLE
ENABLED } ;

1.5.23 FIXED USERNAME Subclause (Server)

Specifies that the user name, UIC, and default directory of the server are those associated with the user name under which the server process starts.

FIXED USERNAME ;

1.5.24 GLOBAL Subclause (Task)

Specifies that a task can be selected from a menu, called by an agent, or called by another task.

GLOBAL ;

1.5.25 LOCAL Subclause (Task)

Specifies that a task can be called by or chained to another task, but not selected from a menu or called by an agent.

LOCAL ;

1.5.26 LOGICALS Subclause (Server)

Defines a set of process logical names for one or more server processes.

$$\left\{ \begin{array}{l} \underline{\text{LOGICAL}} \\ \underline{\text{LOGICALS}} \end{array} \right\} \left[\begin{array}{l} \text{NAME IS} \\ \text{NAMES ARE} \end{array} \right]$$

$$\left\{ \left\{ \begin{array}{l} \text{logical-name} \\ \text{logical-string} \end{array} \right\} = \left\{ \begin{array}{l} \text{equivalence-name} \\ \text{equivalence-string} \end{array} \right\} \right\} [\dots];$$

1.5.27 MAXIMUM SERVER PROCESSES Clause (Application, Server)

Sets the maximum number of OpenVMS processes that can be created for an application or for a particular server within an application. This number cannot exceed the maximum number of processes that can be created for any given system (determined by the SYSGEN parameter MAXPROCESSCNT).

$$\left\{ \begin{array}{l} \underline{\text{MAXIMUM}} \\ \underline{\text{MAX}} \end{array} \right\} \text{SERVER} \left\{ \begin{array}{l} \underline{\text{PROCESS}} \\ \underline{\text{PROCESSES}} \end{array} \right\} \text{IS} \left\{ \begin{array}{l} \text{high-number} \\ \underline{\text{UNLIMITED}} \end{array} \right\};$$

1.5.28 MAXIMUM TASK INSTANCES Clause (Application)

Sets the largest number of task instances that can be active at one time for an application.

$$\left\{ \begin{array}{l} \underline{\text{MAXIMUM}} \\ \underline{\text{MAX}} \end{array} \right\} \text{TASK} \left\{ \begin{array}{l} \underline{\text{INSTANCE}} \\ \underline{\text{INSTANCES}} \end{array} \right\} \text{IS} \left\{ \begin{array}{l} \text{high-number} \\ \underline{\text{UNLIMITED}} \end{array} \right\};$$

1.5.29 MINIMUM SERVER PROCESSES Subclause (Server)

Sets the minimum number of server processes that you want ACMS to have available for a server at one time.

$$\left\{ \begin{array}{l} \underline{\text{MINIMUM}} \\ \underline{\text{MIN}} \end{array} \right\} \text{SERVER} \left\{ \begin{array}{l} \underline{\text{PROCESS}} \\ \underline{\text{PROCESSES}} \end{array} \right\} \text{IS low-number};$$

1.5.30 NAME TABLES Subclause (Server)

Specifies one or more logical name tables the server process can use.

$$\left[\text{NAME} \right] \left\{ \begin{array}{l} \underline{\text{TABLE IS}} \\ \underline{\text{TABLES ARE}} \end{array} \right\} \left\{ \begin{array}{l} \text{logical-name-table} \\ \text{quoted-string} \end{array} \right\} [\dots];$$

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.31 PROTECTED WORKSPACES Subclause (Server)

Enables a workspace mapping option that maps the entire task instance workspace pool during the first procedure call to a task server. The workspaces stay mapped until the server runs down.

```
[ NO ] { PROTECTED } { WORKSPACE } ;  
         { PROTECT } { WORKSPACES } ;
```

1.5.32 SERVER ATTRIBUTES Clause (Application)

Defines the control attributes for individual servers. Both the SERVER ATTRIBUTES and SERVER DEFAULTS clauses use the same subclauses.

```
SERVER CONTROL { ATTRIBUTE IS }  
                { ATTRIBUTES ARE }  
  
    { server-given-name: [SERVER group-server-name] [IN task-group-name] ; } ...  
    { <server-subclause> ... }  
  
END SERVER CONTROL [ ATTRIBUTE ] ;  
                    [ ATTRIBUTES ] ;
```

1.5.33 SERVER DEFAULTS Clause (Application)

Changes one or more of the current default settings for one or more server control attributes. The changes you make with the SERVER DEFAULTS clause affect all of the servers defined explicitly or implicitly after the SERVER DEFAULTS clause.

```
SERVER { DEFAULT IS }  
        { DEFAULTS ARE }  
        <server-subclause> ...  
  
END SERVER [ DEFAULT ] ;  
             [ DEFAULTS ] ;
```

1.5.34 SERVER MONITORING INTERVAL Clause (Application)

Controls how often queues are checked to determine whether or not to create or delete new server processes.

```
SERVER MONITORING INTERVAL IS seconds ;
```

1.5.35 SERVER PROCESS DUMP Subclause (Server)

Specifies whether or not an OpenVMS process dump is generated for a server process if the process terminates abnormally.

```
[ NO ] [ SERVER ] PROCESS { DUMP } ;  
                          { DUMPS } ;
```

1.5.36 TASK ATTRIBUTES Clause (Application)

Defines one or more task control attributes on a task-by-task basis.

```

TASK CONTROL { ATTRIBUTE IS
               ATTRIBUTES ARE }
{ task-given-name: [TASK group-task-name] [IN task-group-name] ; } ...
  [<task-subclause> ... ]
END TASK CONTROL [ ATTRIBUTE
                  ATTRIBUTES ] ;

```

1.5.37 TASK DEFAULTS Clause (Application)

Changes the default values for one or more task control attributes in an application definition.

```

TASK { DEFAULT IS
        DEFAULTS ARE }
      <task-subclause> ...
END TASK [ DEFAULT
          DEFAULTS ] ;

```

1.5.38 TASK GROUPS Clause (Application)

Names the task groups containing the tasks associated with an application. Include at least one TASK GROUPS clause in each application definition you write.

```

TASK { GROUP IS
        GROUPS ARE }
{ task-group-given-name: TASK GROUP FILE IS task-group-file ; } ...
END TASK [ GROUP
          GROUPS ] ;

```

1.5.39 TRANSACTION TIMEOUT Subclause (Task)

Places a limit on how long a distributed transaction can remain active.

```
[ NO ] TRANSACTION TIMEOUT IS seconds;
```

1.5.40 USERNAME Subclause (Server)

Defines the user name the server process runs under.

```

USERNAME IS { username
              USERNAME OF { TERMINAL USER
                           APPLICATION } } ;

```

ADU Commands and Clauses

1.5 Application Definition Clauses

1.5.41 WAIT Subclause (Task)

Controls whether or not ACMS displays a message prompting users to press `[Return]`. Pressing `[Return]` clears the terminal screen and displays the previous ACMS menu.

`[NO] WAIT ;`

1.6 Menu Definition Clauses

Menu definitions describe the contents of ACMS menus, which are screen displays of entries that users can select. Users can select task entries that do the work of an application, or menu entries that display other menus with their own entries. This section lists the syntax for the ADU clauses and subclauses you use to define menus.

The `ENTRIES` clause is the only required menu clause. It includes a required subclause specifying whether an entry selects a task or another menu. The `ENTRIES` clause can also include optional subclauses for displaying descriptive text and controlling screen display characteristics.

This section begins with overview syntax for menu definitions. The overview syntax is followed by syntax for individual menu definition clauses and subclauses and for the application specification parameter.

1.6.1 Menu Definition Syntax

[DEFAULT APPLICATION IS application-spec ;]

[DEFAULT MENU FILE IS menu-database-file ;]

[HEADER IS string [,string] ;]

{ REQUEST IS request-name [WITH number ENTRIES PER SCREEN] ; }

{ [SEND] CONTROL TEXT [IS] { string
quoted-string }
[WITH number ENTRIES PER SCREEN] ; }

{ { ENTRY IS
ENTRIES ARE } }

entry-name:
{ { MENU IS menu-path-name ;
TASK IS task-name [IN application-spec] ; } }
{ [TEXT IS description-string ;] } ...
{ [[NO] DELAY]
[[NO] WAIT] } ...

END [ENTRY
ENTRIES] ;

END DEFINITION ;

ADU Commands and Clauses

1.6 Menu Definition Clauses

1.6.2 CONTROL TEXT Clause (Menu)

Lets you customize your HP DECforms menu by sending up to five control text items to the form.

$$\left[\begin{array}{l} [\text{SEND}] \text{CONTROL TEXT} [\text{IS}] \left\{ \begin{array}{l} \text{string} \\ \text{quoted-string} \end{array} \right\} \\ [\text{WITH number ENTRIES PER SCREEN}]; \end{array} \right]$$

1.6.3 DEFAULT APPLICATION Clause (Menu)

Defines the application specification that ACMS uses as the default for TASK entries, unless you name a different application database file with the TASK subclause.

DEFAULT APPLICATION IS application-spec ;

1.6.4 DEFAULT MENU FILE Clause (Menu)

Defines the menu database file that ACMS uses as the default when it builds a menu tree. ACMS builds the menu tree when you use the ADU BUILD command and includes the specified menu as the top menu in the tree.

DEFAULT MENU FILE IS menu-database-file ;

1.6.5 DELAY Subclause (Optional ENTRIES)

Controls whether or not ACMS waits 3 seconds after a task entry stops running before clearing the screen and displaying the ACMS menu.

[NO] DELAY ;

1.6.6 ENTRIES Clause (Menu)

Defines an entry as a menu entry or as a task entry. A menu entry displays a menu when users select the entry. A task entry runs a task.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{ENTRY IS} \\ \text{ENTRIES ARE} \end{array} \right\} \\ \\ \left\{ \begin{array}{l} \text{entry-name:} <\text{required-subclause}> \dots \\ <\text{optional-subclause}> \dots \end{array} \right\} \dots \\ \\ \text{END} [\text{ENTRY} \\ \text{ENTRIES}] ; \end{array} \right\}$$

1.6.7 HEADER Clause (Menu)

Defines the title of a menu.

HEADER IS string [,string] ;

1.6.8 MENU Subclause (Required ENTRIES)

Defines an entry as a menu and points to the CDD location of the definition for that menu.

MENU IS menu-path-name ;

1.6.9 REQUEST Clause (Menu)

Identifies the TDMS request that defines the menu layout.

REQUEST IS request-name [WITH number ENTRIES PER SCREEN] ;

1.6.10 TASK Subclause (Required ENTRIES)

Names the task ACMS runs when a user selects the entry from a menu.

TASK IS task-name [IN application-spec] ;

1.6.11 TEXT Subclause (Optional ENTRIES)

Provides descriptive text that ACMS displays with a menu or task entry.

TEXT IS description-string ;

1.6.12 WAIT Subclause (Optional ENTRIES)

Controls whether or not ACMS prompts a user to press Return, after a task entry completes, before clearing the screen and displaying the ACMS menu.

[NO] WAIT ;

1.6.13 Application Specification Parameter

Several ADU clauses include an application specification parameter. For example, in the DEFAULT APPLICATION clause, you must specify the name of the application that is the default when the user signs in to ACMS.

The syntax for the application specification parameter is:

$$["] \left\{ \begin{array}{l} \left[\left\{ \begin{array}{l} \text{node-name} \\ \text{logical-node-name} \end{array} \right\} :: \right] \text{file-name} \\ \text{logical-appl-name} \end{array} \right\} ["]$$

ADU Commands and Clauses

1.7 Declining Features Syntax

1.7 Declining Features Syntax

The task clauses and phrases in this section are considered to be declining features in the task definition language. It is recommended that you use the distributed transaction syntax to control file and database transactions. Most of the clauses and phrases in this section are for declaring file and database recovery units in the task definition.

In addition to clauses and phrases related to file and database recovery units, this section contains the `CONTINUE ON BAD STATUS` phrase and the `GOTO TASK` and `REPEAT TASK` clauses. It is recommended that you use the `RAISE EXCEPTION` and `EXCEPTION HANDLER` clauses instead of `CONTINUE ON BAD STATUS`; the `CALL TASK` clause instead of `GOTO TASK`; and `REPEAT STEP` instead of `REPEAT TASK`.

Existing applications that use the clauses and phrases in this section can run under ACMS Version 4.0 or higher.

1.7.1 COMMIT Clause (Action)

Signals the end of the current transaction in steps you define using DBMS, Rdb, RMS, or SQL recovery units. Also causes any changes made since the start of the transaction to be written to the DBMS or Rdb database or an RMS file.

```
COMMIT [ RETAINING RECOVERY UNIT ] [ IF ACTIVE RECOVERY UNIT ] ;
```

1.7.2 CONTINUE ON BAD STATUS Phrase (Processing)

Instructs ACMS to continue task execution if a task called by a processing step returns a failure status.

```
CONTINUE ON BAD STATUS
```

1.7.3 DBMS RECOVERY Phrase (Block, Processing)

The `DBMS RECOVERY` phrase readies a DBMS database at the start of a block or processing step.

```
DBMS RECOVERY dml-string [...]
```

1.7.4 GOTO TASK Clause (Action)

Ends the current task and starts a new task without requiring the user to make a task selection.

```
[ GO TO  
GOTO ] TASK task-name [ PASSING workspace-name [ ,... ] ] ;
```

1.7.5 NO RECOVERY UNIT ACTION Clause (Action)

Specifies that there is no action taken on any active recovery unit.

```
NO RECOVERY UNIT ACTION ;
```

1.7.6 RDB RECOVERY Phrase (Block, Processing)

Starts an Rdb database transaction at the beginning of a block or processing step.

RDB RECOVERY dml-string [...]

1.7.7 REPEAT TASK Clause (Action)

Ends the current task instance and restarts the task without requiring the user to select the task from a menu.

REPEAT TASK [PASSING workspace-name [...]] ;

1.7.8 RETAIN RECOVERY UNIT Clause (Action)

Maintains the recovery unit within the current server.

RETAIN RECOVERY UNIT [IF ACTIVE RECOVERY UNIT] ;

1.7.9 RMS RECOVERY Phrase (Block, Processing)

Starts an RMS recovery unit for a block or processing step.

RMS RECOVERY

1.7.10 ROLLBACK Clause (Action)

Signals the end of a recovery unit, returning all recoverable objects to the state they were in at the beginning of the current recovery unit.

ROLLBACK [IF ACTIVE RECOVERY UNIT] ;

1.7.11 SQL RECOVERY Phrase (Block, Processing)

Starts an SQL transaction with an Rdb database or a VIDA database at the beginning of a block or processing step.

SQL RECOVERY dml-string [...]

ACMS Management Utilities and Commands

This chapter contains syntax for the ACMS application and system management utilities, and operator and show commands.

2.1 ACMSQUEMGR Commands

This section contains syntax for the ACMS Queue Manager (ACMSQUEMGR) Utility commands. ACMSQUEMGR commands allow you to create and manage ACMS task queues and the queued task elements in the queues. See *HP ACMS for OpenVMS Managing Applications* for more information about ACMSQUEMGR commands and qualifiers.

2.1.1 CREATE QUEUE Command (ACMSQUEMGR>)

Creates a queue for queued task elements.

```
CREATE QUEUE queue-name
```

Command Qualifiers	Defaults
/DEQUEUE=keyword	/DEQUEUE=RESUMED
/ENQUEUE=keyword	/ENQUEUE=RESUMED
/FILE_SPECIFICATION=file-spec	SYS\$SYSTEM:<queue-name>.DAT
/MAX_WORKSPACES_SIZE=n	/MAX_WORKSPACES_SIZE=256

2.1.2 DELETE ELEMENT Command (ACMSQUEMGR>)

Removes one or more queued task elements from the specified queue.

```
DELETE ELEMENT element-id queue-name
```

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/EXCLUDE=(keyword[,...])	None
/SELECT=(keyword[,...])	None

2.1.3 DELETE QUEUE Command (ACMSQUEMGR>)

Deletes the queue you specify.

```
DELETE QUEUE queue-name
```

Command Qualifiers	Defaults
/[NO]PURGE	/NOPURGE

ACMS Management Utilities and Commands

2.1 ACMSQUEMGR Commands

2.1.4 EXIT Command (ACMSQUEMGR>)

Ends the ACMSQUEMGR session and returns to DCL level.

EXIT

2.1.5 HELP Command (ACMSQUEMGR>)

Displays information about ACMSQUEMGR commands and qualifiers.

HELP [topic [...]]

Command Qualifiers	Defaults
/[NO]PROMPT	/PROMPT

2.1.6 MODIFY QUEUE Command (ACMSQUEMGR>)

With a qualifier, modifies the static characteristics of a task queue.

MODIFY QUEUE queue-name

Command Qualifiers	Defaults
/FILE_SPECIFICATION=file-spec	Existing queue repository file
/MAX_WORKSPACES_SIZE=n	Existing definition specification

2.1.7 SET ELEMENT Command (ACMSQUEMGR>)

With the /PRIORITY qualifier, sets the priority of one or more queued task elements. With the /STATE qualifier, sets the state of one or more queued task elements.

SET ELEMENT element-id queue-name

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/EXCLUDE=(keyword[,...])	None
/PRIORITY=n	None
/SELECT=(keyword[,...])	None
/STATE=[NO]HOLD	None

2.1.8 SET QUEUE Command (ACMSQUEMGR>)

With a qualifier, dynamically sets the queue state. The changes to the queue state take effect immediately.

SET QUEUE queue-name

Command Qualifiers	Defaults
/DEQUEUE=keyword	Current queue state
/ENQUEUE=keyword	Current queue state

2.1.9 SHOW ELEMENT Command (ACMSQUEMGR>)

Displays information about one or more queued task elements in a queue.

SHOW ELEMENT element-id queue-name

Command Qualifiers	Defaults
/BRIEF	/BRIEF
/EXCLUDE=(keyword[,...])	None
/FULL	/BRIEF
/OUTPUT[=file-spec]	/OUTPUT=SYS\$OUTPUT
/SELECT=(keyword[,...])	None
/TOTAL_ONLY	/BRIEF

2.1.10 SHOW QUEUE Command (ACMSQUEMGR>)

Displays the characteristics of the task queue you specify.

SHOW QUEUE queue-name

Command Qualifiers	Defaults
/OUTPUT[=file-spec]	/OUTPUT=SYS\$OUTPUT

2.2 AAU Commands

This section contains syntax for the ACMS Application Authorization Utility (AAU) commands. AAU commands allow you to authorize ACMS applications. See *HP ACMS for OpenVMS Managing Applications* for more information on AAU commands and qualifiers.

2.2.1 ADD Command (AAU>)

Authorizes one or more application names for installation in ACMS\$DIRECTORY.

ADD application-name

Command Qualifiers	Defaults
/ACL=(access-control-list[,...])	From DEFAULT definition
/APPL_USERNAME=username	From DEFAULT definition
/[NO]DYNAMIC_USERNAMES	From DEFAULT definition
/SRV_USERNAMES[(server-username[,...])]	From DEFAULT definition
/[NO]WILD_SUFFIX	From DEFAULT definition

2.2.2 COPY Command (AAU>)

Makes a copy of an existing application authorization.

COPY source-application-name new-application-name

Command Qualifiers	Defaults
/ACL=(access-control-list[,...])	From source authorization
/APPL_USERNAME=username	From source authorization
/[NO]DYNAMIC_USERNAMES	From source authorization

ACMS Management Utilities and Commands

2.2 AAU Commands

/SRV_USERNAMES[=(server-username[,...])]	From source authorization
/[NO]WILD_SUFFIX	From source authorization

2.2.3 DEFAULT Command (AAU>)

Changes information in the DEFAULT authorization.

DEFAULT

Command Qualifiers	Defaults
/ACL=(access-control-list[,...])	From existing authorization
/APPL_USERNAME=username	From existing authorization
/[NO]DYNAMIC_USERNAMES	From existing authorization
/SRV_USERNAMES[=(server-username[,...])]	From existing authorization
/[NO]WILD_SUFFIX	From existing authorization

2.2.4 EXIT Command (AAU>)

Ends an AAU session and returns to the DCL prompt.

EXIT

2.2.5 HELP Command (AAU>)

Displays information about AAU commands and qualifiers.

HELP [topic [...]]

Command Qualifiers	Defaults
/[NO]PROMPT	/PROMPT

2.2.6 LIST Command (AAU>)

Writes the contents of an authorization to ACMSAAU.LIS in your default directory or to an output file you specify.

LIST application-name

Command Qualifiers	Defaults
/BRIEF	Full authorizations
/OUTPUT[=file-spec]	/OUTPUT=ACMSAAU.LIS

2.2.7 MODIFY Command (AAU>)

Changes information in an application authorization.

MODIFY application-name

Command Qualifiers	Defaults
/ACL=(access-control-list[,...])	From existing authorization
/APPL_USERNAME=username	From existing authorization
/[NO]DYNAMIC_USERNAMES	From existing authorization
/SRV_USERNAMES[=(server-username[,...])]	From existing authorization
/[NO]WILD_SUFFIX	From existing authorization

2.2.8 REMOVE Command (AAU>)

Deletes an authorization from the ACMSAAF.DAT application authorization database file.

REMOVE application-name

2.2.9 RENAME Command (AAU>)

Gives an application authorization a new name.

RENAME old-application-name new-application-name

Command Qualifiers

/ACL=(access-control-list[,...])
/APPL_USERNAME=username
/[NO]DYNAMIC_USERNAMES
/SRV_USERNAMES=(server-username[,...])
/[NO]WILD_SUFFIX

Defaults

From old authorization
From old authorization
From old authorization
From old authorization
From old authorization

2.2.10 SHOW Command (AAU>)

Displays information about application authorizations on your terminal screen.

SHOW application-name

Command Qualifiers

/BRIEF

Defaults

Displays full authorizations

2.3 ACMSGEN Commands

This section contains syntax for the ACMSGEN Utility commands. ACMSGEN commands allow you to modify ACMS parameters. See *HP ACMS for OpenVMS Managing Applications* for more information about the ACMSGEN commands and qualifiers.

2.3.1 EXIT Command (ACMSGEN>)

Ends the ACMSGEN session and returns to the DCL prompt.

EXIT

2.3.2 HELP Command (ACMSGEN>)

Displays information about ACMSGEN commands and qualifiers.

HELP [topic [...]]

ACMS Management Utilities and Commands

2.3 ACMSGEN Commands

2.3.3 SET Command (ACMSGEN>)

Changes parameter values in the ACMSGEN work area. The parameter changes are not made to any real parameter until you use the WRITE command.

SET parameter-name value

2.3.4 SHOW Command (ACMSGEN>)

Displays the value in the work area, the default value, the minimum value, the maximum value, the unit of measure, and the dynamic/fixed status for ACMS system parameters.

```
SHOW { parameter-name }  
     { /qualifier [...] }
```

Command Qualifiers	Defaults
/ACC	None
/ALL	None
/CP	None
/EXC	None
/MSS	None
/QTI	None
/TSC	None

2.3.5 USE Command (ACMSGEN>)

Initializes the ACMSGEN work area with values from a work file.

USE file-spec

2.3.6 USE ACTIVE Command (ACMSGEN>)

Initializes the ACMSGEN work area with active values for all parameters from an ACMS system global section.

USE ACTIVE

2.3.7 USE CURRENT Command (ACMSGEN>)

Initializes the ACMSGEN work area with current values for all parameters from the SYS\$SYSTEM:ACMSPAR.ACM parameter file.

USE CURRENT

2.3.8 USE DEFAULT Command (ACMSGEN>)

Initializes the ACMSGEN work area with ACMS default values for all ACMS parameters.

USE DEFAULT

2.3.9 WRITE Command (ACMSGEN>)

Writes values from the ACMSGEN work area to a work file, creating a new version of the file.

WRITE file-spec

2.3.10 WRITE ACTIVE Command (ACMSGEN>)

Changes active values for dynamic parameters by writing values from the ACMSGEN work area to an ACMS system global section.

WRITE ACTIVE

2.3.11 WRITE CURRENT Command (ACMSGEN>)

Changes current values by writing values from the ACMSGEN work area to the SYS\$SYSTEM:ACMSPAR.ACM file.

WRITE CURRENT

2.4 ATR Commands

This section contains syntax for the ACMS Audit Trail Report (ATR) Utility commands. ATR commands allow you to generate reports containing information logged by the Audit Trail Logger. See *HP ACMS for OpenVMS Managing Applications* for more information on ATR commands and qualifiers.

2.4.1 EXIT Command (ATR>)

Ends the ATR Utility session and returns to the DCL prompt.

EXIT

2.4.2 HELP Command (ATR>)

Displays information about ATR Utility commands and their qualifiers.

HELP [topic [...]]

2.4.3 LIST Command (ATR>)

Produces a report about information in the Audit Trail Log file. You can limit the amount of information in the report by using qualifiers.

LIST [file-spec]

Command Qualifiers	Defaults
/APPLICATION=application-name	All application names
/BEFORE[=time]	Full report
/BRIEF	Full report
/IDENTIFICATION=task-id	All task IDs
/OUTPUT=file-spec	/OUTPUT=SYS\$OUTPUT

ACMS Management Utilities and Commands

2.4 ATR Commands

/SINCE[=time]	Full report
/SUBMITTER=submitter-id	All submitter IDs
/TASK=task-name	All task names
/TERMINAL=device-name	All device names
/TYPE=type	All types
/USERNAME=user-name	All user names

2.5 DDU Commands

This section contains syntax for the ACMS Device Definition Utility (DDU) commands. You can use DDU commands to authorize and control ACMS terminals. See *HP ACMS for OpenVMS Managing Applications* for more information about DDU commands and qualifiers.

2.5.1 ADD Command (DDU>)

Authorizes and assigns login characteristics to an ACMS terminal by creating a DDU definition and adding it to the device authorization file (ACMSDDF.DAT).

ADD device-name

Command Qualifiers	Defaults
/[NO]AUTOLOGIN=username	From DEFAULT definition
/[NO]CONTROLLED	From DEFAULT definition
/PRINTFILE [= print-file-spec spooled-device-name]	From DEFAULT definition

2.5.2 COPY Command (DDU>)

Authorizes a terminal, using the login characteristics from the DDU definition you specify.

COPY source-device-name new-device-name

Command Qualifiers	Defaults
/[NO]AUTOLOGIN=username	From source definition
/[NO]CONTROLLED	From source definition
/PRINTFILE [= print-file-spec spooled-device-name]	From source definition

2.5.3 DEFAULT Command (DDU>)

Changes information in the DDU DEFAULT definition.

DEFAULT

Command Qualifiers	Defaults
/[NO]AUTOLOGIN=username	From current DEFAULT definition
/[NO]CONTROLLED	From current DEFAULT definition
/PRINTFILE [= print-file-spec spooled-device-name]	From current DEFAULT definition

2.5.4 EXIT Command (DDU>)

Ends the DDU session and returns to the DCL prompt.

EXIT

2.5.5 HELP Command (DDU>)

Displays information about DDU commands and qualifiers.

HELP [topic [...]]

Command Qualifiers	Defaults
/[NO]PROMPT	/PROMPT

2.5.6 LIST Command (DDU>)

Writes DDU definitions to ACMSDDU.LIS in your default directory, or to an output file you specify.

LIST device-name

Command Qualifiers	Defaults
/BRIEF	Full DDU definitions
/OUTPUT[=file-spec]	/OUTPUT=ACMSDDU.LIS

2.5.7 MODIFY Command (DDU>)

Changes the login characteristics in a DDU definition.

MODIFY device-name

Command Qualifiers	Defaults		
/[NO]AUTOLOGIN=username	From current definition		
/[NO]CONTROLLED	From current definition		
/PRINTFILE [= <table border="0" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 0 5px;">print-file-spec</td> <td style="padding: 0 5px;">spooled-device-name</td> </tr> </table>]	print-file-spec	spooled-device-name	From current definition
print-file-spec	spooled-device-name		

2.5.8 REMOVE Command (DDU>)

Removes a DDU definition from the device authorization file.

REMOVE device-name

2.5.9 RENAME Command (DDU>)

Changes the device name in a DDU definition.

RENAME old-device-name new-device-name

Command Qualifiers	Defaults
/[NO]AUTOLOGIN=username	From old definition
/[NO]CONTROLLED	From old definition

ACMS Management Utilities and Commands

2.5 DDU Commands

`/PRINTFILE` [= print-file-spec
spooled-device-name] From old definition

2.5.10 SHOW Command (DDU>)

Displays DDU definitions.

SHOW device-name

Command Qualifiers	Defaults
/BRIEF	Full definition

2.6 Operator Commands

This section contains syntax for the ACMS operator commands. ACMS operator commands allow you to control the ACMS system and its components. See *HP ACMS for OpenVMS Managing Applications* for more information about ACMS operator commands and qualifiers.

2.6.1 ACMS/CANCEL TASK Command

Stops one or more task instances. With qualifiers, stops only the task instance you identify.

ACMS/CANCEL TASK [task-name]

Command Qualifiers	Defaults
/APPLICATION=application-name	All applications
/[NO]CONFIRM	/CONFIRM
/DEVICE=device-name	All devices
/IDENTIFIER=task-id	All tasks
/[NO]LOG	/NOLOG
/SUBMITTER=submitter-id	All submitters
/USER=user-name	All user names

2.6.2 ACMS/CANCEL USER Command

Cancels a user by stopping all of the user's outstanding tasks and by signing the user out of ACMS.

ACMS/CANCEL USER [user-name]

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/DEVICE=device-name	All devices
/[NO]LOG	/NOLOG
/SUBMITTER=submitter-id	All submitters

2.6.3 ACMS/DEBUG Command

Starts the ACMS Task Debugger. This command allows you to test tasks and server procedures without building an entire ACMS application.

ACMS/DEBUG [task-group name]

Command Qualifiers	Defaults
/AGENT_HANDLE	None
/PID	None
/SERVER	None
/TWS_POOLSIZ[=n]	1600 pagelets (on Alpha and I64)
/TWSC_POOLSIZ[=n]	50 pagelets (on Alpha and I64)
/WORKSPACE	None

2.6.4 ACMS/ENTER Command

Allows a terminal that has logged in to OpenVMS to use the ACMS menu system.

ACMS/ENTER

Command Qualifiers	Defaults
/[NO]RETURN	/RETURN

2.6.5 ACMS/INSTALL Command

Installs an application in ACMS\$DIRECTORY or removes an application database file from ACMS\$DIRECTORY.

ACMS/INSTALL file-spec

Command Qualifiers	Defaults
/[NO]REMOVE	/NOREMOVE

2.6.6 ACMS/MODIFY APPLICATION Command

Modifies the attributes of an active application.

ACMS/MODIFY APPLICATION [application-name[,...]]

Command Qualifiers	Defaults
/APPLICATION_ATTRIBUTES=(attribute[,...])	None
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/NOLOG
/SERVER_ATTRIBUTES=(attribute[,...])	None
/TASK_ATTRIBUTES=(attribute[,...])	None

ACMS Management Utilities and Commands

2.6 Operator Commands

2.6.7 ACMS/REPLACE SERVER Command

Replaces a server image with a new version of that image. All subsequent tasks use the new image.

ACMS/REPLACE SERVER [server-name[,...]]

Command Qualifiers	Defaults
/APPLICATION=application-name	All applications
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/NOLOG

2.6.8 ACMS/REPROCESS APPLICATION_SPEC Command

Causes ACMS to retranslate the application specification for an application and redirect all subsequent task selections to the application pointed to by the application specification.

ACMS/REPROCESS APPLICATION_SPEC application-spec

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/[NO]LOG	/NOLOG

2.6.9 ACMS/RESET AUDIT Command

Resets the ACMS Audit Trail Log, and causes the current ACMS Audit Trail Log file to close and a new log file to open.

ACMS/RESET AUDIT

2.6.10 ACMS/RESET TERMINALS Command

Causes ACMS to read the Device Definition Utility (DDU) database file, authorize any new controlled terminals, and release any terminals no longer authorized.

ACMS/RESET TERMINALS

2.6.11 ACMS/SET QUEUE Command

Sets the processing characteristics of a started task queue. The processing characteristics are used by the queued task initiator (QTI) to process tasks in a queue.

ACMS/SET QUEUE queue-name[,...]

Command Qualifiers	Defaults
/TASK_THREADS=n	/TASK_THREADS=1

ACMS Management Utilities and Commands

2.6 Operator Commands

2.6.12 ACMS/SET SYSTEM Command

Depending on the qualifiers used, enables or disables Audit Trail logging, or enables or disables ACMS operator terminals.

ACMS/SET SYSTEM

Command Qualifiers	Defaults
/[NO]AUDIT	Current setting
/[NO]OPERATOR	Current setting
/PROCESS	Current setting
/TERMINAL=device-name	Current setting

2.6.13 ACMS/SHOW APPLICATION Command

Displays information about one or more active ACMS applications in static mode. See the ACMS/SHOW APPLICATION/CONTINUOUS command to display application information in continuous mode.

ACMS/SHOW APPLICATION [application-name[,...]]

Command Qualifiers	Defaults
/CONNECTIONS	No connection information displayed
/DETACHED_TASKS	No detached task information displayed
/POOL	No pool information displayed
/SERVER_ATTRIBUTES	No server attributes displayed
/TASK_ATTRIBUTES	No task attributes displayed

2.6.14 ACMS/SHOW APPLICATION/CONTINUOUS Command

Displays information about an active ACMS application in continuous refresh mode.

ACMS/SHOW APPLICATION/CONTINUOUS application-name

Command Qualifiers	Defaults
/[NO]BEGINNING_TIME=time	/NOBEGINNING_TIME
/[NO]ENDING_TIME=time	/NOENDING_TIME
/[NO]INTERVAL[=seconds]	/NOINTERVAL
/[NO]OUTPUT[=file-spec]	/NOOUTPUT

2.6.15 ACMS/SHOW QTI Command

Displays the run-time characteristics of the queued task initiator (QTI).

ACMS/SHOW QTI

2.6.16 ACMS/SHOW QUEUE Command

Displays information about one or more task queues.

ACMS/SHOW QUEUE [queue-name[,...]]

ACMS Management Utilities and Commands

2.6 Operator Commands

2.6.17 ACMS/SHOW SERVER Command

Displays information about one or more servers running under a specified application.

ACMS/SHOW SERVER [server-name [...]]

Command Qualifiers	Defaults
/APPLICATION=application-name	All applications

2.6.18 ACMS/SHOW SYSTEM Command

Displays information about the ACMS run-time system, all local and remote users, and all local applications.

ACMS/SHOW SYSTEM

Command Qualifiers	Defaults
/POOL	No pool information displayed
/ALL	All processes using message-switch displayed

2.6.19 ACMS/SHOW TASK Command

Displays information about one or more active ACMS tasks executing on the local node.

ACMS/SHOW TASK [task-name[...]]

Command Qualifiers	Defaults
/APPLICATION=application-name	All applications
/DEVICE=device-name	All devices
/IDENTIFIER=task-id	All tasks
/SUBMITTER=submitter-id	All submitters
/USER=user-name	All users

2.6.20 ACMS/SHOW USER Command

Displays information about ACMS users. With qualifiers, displays information about only those users you identify.

ACMS/SHOW USER [user-name[...]]

Command Qualifiers	Defaults
/ALL	/ALL
/APPLICATION	All applications
/DEVICE=device-name	All devices
/[NO]FULL	/NOFULL
/LOCAL	All submitters
/REMOTE	All submitters
/SUBMITTER=submitter-id	All submitters

2.6.21 ACMS/START APPLICATION Command

Starts one or more ACMS applications.

ACMS/START APPLICATION application-name[,...]

2.6.22 ACMS/START QTI Command

Starts the queued task initiator (QTI).

ACMS/START QTI

2.6.23 ACMS/START QUEUE Command

Starts the task queue you specify. Once you start a task queue, the QTI begins processing any queued task elements in the queue.

ACMS/START QUEUE queue-name[,...]

Command Qualifiers

/ERROR_QUEUE=error-queue-name
/TASK_THREADS=n

Defaults

No error queue
/TASK_THREADS=1

2.6.24 ACMS/START SYSTEM Command

Starts the ACMS system.

ACMS/START SYSTEM

Command Qualifiers

/[NO]AUDIT
/[NO]QTI
/[NO]TERMINALS

Defaults

/AUDIT
/NOQTI
/TERMINALS

2.6.25 ACMS/START TASK Command

Starts a detached task in the specified application.

ACMS/START TASK task-name application-name

Command Qualifiers

/[NO]LOG
/[NO]RETRY_LIMIT[=n]
/SELECTION_STRING=selection_string
/USERNAME=username
/WAIT_TIMER=n

Defaults

/NOLOG
/RETRY_LIMIT=0
Null string
User name of application
/WAIT_TIMER=5 seconds

2.6.26 ACMS/START TERMINALS Command

Starts the terminal subsystem controller (TSC) when ACMS is running. The ACMS/START TERMINALS command enables terminal users to access ACMS menus.

ACMS/START TERMINALS

ACMS Management Utilities and Commands

2.6 Operator Commands

2.6.27 ACMS/STOP APPLICATION Command

Stops one or more ACMS applications.

ACMS/STOP APPLICATION application-name[,...]

Command Qualifiers	Defaults
/[NO]CANCEL	/NOCANCEL

2.6.28 ACMS/STOP QTI Command

Stops the queued task initiator (QTI) and all active task queues.

ACMS/STOP QTI

Command Qualifiers	Defaults
/[NO]CANCEL	/NOCANCEL

2.6.29 ACMS/STOP QUEUE Command

Stops the specified task queue.

ACMS/STOP QUEUE queue-name[,...]

Command Qualifiers	Defaults
/[NO]CANCEL	/NOCANCEL

2.6.30 ACMS/STOP SYSTEM Command

Stops the ACMS system, the terminal subsystem controller (TSC), the queued task initiator (QTI), and all applications.

ACMS/STOP SYSTEM

Command Qualifiers	Defaults
/[NO]CANCEL	/NOCANCEL

2.6.31 ACMS/STOP TERMINALS Command

Stops the terminal subsystem controller (TSC), thereby canceling the tasks of all ACMS menu users and logging out all current terminal users.

ACMS/STOP TERMINALS

2.7 SWLUP Commands

This section contains syntax for the ACMS Software Event Log Utility (SWLUP) commands. SWLUP commands allow you to generate reports containing information logged by the Software Event Logger (SWL). See *HP ACMS for OpenVMS Managing Applications* for more information about SWLUP commands and qualifiers.

2.7.1 @ (At sign) Command (SWLUP>)

Runs an indirect command file that contains SWLUP commands.

@ file-spec

2.7.2 EDIT Command (SWLUP>)

Lets you edit the last SWLUP command you typed, or lets you create an edit buffer for entering SWLUP commands.

EDIT

2.7.3 EXIT Command (SWLUP>)

Causes SWLUP to exit or ends the execution of a command file.

EXIT

2.7.4 HELP Command (SWLUP>)

Displays information about SWLUP commands and qualifiers.

HELP [topic [...]]

Command Qualifiers	Defaults
/[NO]PROMPT	/PROMPT

2.7.5 LIST Command (SWLUP>)

Lists events recorded in the Software Event Logger file.

LIST [EVENTS]

Command Qualifiers	Defaults
/BEFORE[=time]	Full report
/EVENT_CODE=event-code[,...]	All event codes
/FACILITY=facility-name[,...]	All facilities
/IMAGE=image-name[,...]	All images
/INPUT=file-spec	/INPUT=SYS\$ERRORLOG:SWL.LOG
/OUTPUT=file-spec	/OUTPUT=SYS\$OUTPUT
/PRINT	Does not print
/PROCESS_NAME=process-name[,...]	All processes
/SEVERITY=severity-code[,...]	All severity codes
/SINCE[=time]	Full report
/USER=user-name[,...]	All user names

2.7.6 RENEW Command (SWLUP>)

Starts a new system wide Software Event Logger file.

RENEW

ACMS Management Utilities and Commands

2.7 SWLUP Commands

2.7.7 SAVE Command (SWLUP>)

Writes to a file that the last command typed.

SAVE file-spec

2.7.8 SET [NO]LOG Command (SWLUP>)

Enables or disables creation of a log file that records your SWLUP session.

SET LOG [file-spec]

SET NOLOG

2.7.9 SET [NO]VERIFY Command (SWLUP>)

Enables or disables the printing of commands stored in an indirect command file. SWLUP sends output to the default output device SYS\$OUTPUT.

SET [NO]VERIFY

2.7.10 SHOW CURRENT Command (SWLUP>)

Displays the name of the current log file opened by the SWL detached process.

SHOW CURRENT

2.7.11 SHOW LOG Command (SWLUP>)

Displays whether or not you are currently logging SWLUP commands and the name of the log file, if applicable.

SHOW LOG

2.7.12 SHOW VERSION Command (SWLUP>)

Displays the current version of SWLUP on the default output device SYS\$OUTPUT.

SHOW VERSION

2.7.13 STOP Command (SWLUP>)

Stops the SWL detached process so that it can exit properly.

STOP

2.8 UDU Commands

This section contains syntax for the ACMS User Definition Utility (UDU) commands. You can use UDU commands to authorize ACMS users. See *HP ACMS for OpenVMS Managing Applications* for more information about the UDU commands and qualifiers.

2.8.1 ADD Command (UDU>)

Authorizes and assigns sign-in characteristics to ACMS users by adding UDU definitions to the user authorization file (ACMSUDF.DAT). You can use qualifiers to assign sign-in characteristics or let new definitions receive information from the UDU DEFAULT definition.

ADD user-name

Command Qualifiers	Defaults
/[NO]AGENT	From DEFAULT definition
/[NO]DISPLAY_MENU	From DEFAULT definition
/[NO]FINAL=(keyword[...])	From DEFAULT definition
/[NO]INITIAL=(keyword[...])	From DEFAULT definition
/LANGUAGE=language-name	From DEFAULT definition
/MDB=menu-database-file	From DEFAULT definition
/MENU[=menu-path-name]	From DEFAULT definition
/PRINTFILE $\left[\begin{array}{l} = \text{print-file-spec} \\ \text{spooled-device-name} \end{array} \right]$	From DEFAULT definition
/[NO]SKIPMENULANGUAGE	From DEFAULT definition

2.8.2 ADD/PROXY Command (UDU>)

Adds a user proxy to the ACMS proxy file (ACMSPROXY.DAT). Before you can use the ADD /PROXY command, you must have already created a proxy file using the CREATE /PROXY command.

ADD /PROXY remote-node::remote-user local-user

2.8.3 COPY Command (UDU>)

Authorizes and assigns sign-in characteristics to ACMS users by creating new UDU definitions from existing UDU definitions. With qualifiers, you can assign different sign-in characteristics to the new definitions.

COPY source-user-name new-user-name

Command Qualifiers	Defaults
/[NO]AGENT	From source definition
/[NO]DISPLAY_MENU	From source definition
/[NO]FINAL=(keyword[...])	From source definition
/[NO]INITIAL=(keyword[...])	From source definition
/LANGUAGE[=language-name]	From source definition
/MDB=menu-database-file	From source definition
/MENU[=menu-path-name]	From source definition

ACMS Management Utilities and Commands

2.8 UDU Commands

<code>/PRINTFILE</code>	$\left[\begin{array}{l} = \text{print-file-spec} \\ \text{spooled-device-name} \end{array} \right]$	From source definition
<code>/[NO]SKIPMENULANGUAGE</code>		From source definition

2.8.4 CREATE/PROXY Command (UDU>)

Creates an empty ACMS proxy file (ACMSPROXY.DAT).

`CREATE /PROXY`

2.8.5 DEFAULT Command (UDU>)

Changes information in the UDU DEFAULT definition. If you omit one or more qualifiers from an ADD command, the resulting new definition receives information from the existing DEFAULT definition.

`DEFAULT`

Command Qualifiers	Defaults	
<code>/[NO]AGENT</code>	From existing DEFAULT definition	
<code>/[NO]DISPLAY_MENU</code>	From existing DEFAULT definition	
<code>/[NO]FINAL=(keyword[,...])</code>	From existing DEFAULT definition	
<code>/[NO]INITIAL=(keyword[,...])</code>	From existing DEFAULT definition	
<code>/LANGUAGE[=language-name]</code>	From existing DEFAULT definition	
<code>/MDB=menu-database-file</code>	From existing DEFAULT definition	
<code>/MENU[=menu-path-name]</code>	From existing DEFAULT definition	
<code>/PRINTFILE</code>	$\left[\begin{array}{l} = \text{print-file-spec} \\ \text{spooled-device-name} \end{array} \right]$	From existing DEFAULT definition
<code>/[NO]SKIPMENULANGUAGE</code>	From existing DEFAULT definition	

2.8.6 EXIT Command (UDU>)

Ends the UDU session and returns to DCL level.

`EXIT`

2.8.7 HELP Command (UDU>)

Displays information about UDU commands and qualifiers.

`HELP [topic [...]]`

Command Qualifiers	Defaults
<code>/[NO]PROMPT</code>	<code>/PROMPT</code>

2.8.8 LIST Command (UDU>)

Writes UDU definitions to ACMSUDU.LIS in your default directory or to a specified output file.

LIST user-name

Command Qualifiers	Defaults
/BRIEF	Full definition
/OUTPUT[=file-spec]	/OUTPUT=ACMSUDU.LIS

2.8.9 LIST/PROXY Command (UDU>)

Writes all the proxies in the ACMS proxy file to the output file ACMSPROXY.LIS. You can use the /OUTPUT qualifier to specify a different output file name.

LIST /PROXY

Command Qualifier	Default
/OUTPUT=file-spec	ACMSPROXY.LIS

2.8.10 MODIFY Command (UDU>)

Changes the user information by allowing you to change information in UDU definitions.

MODIFY user-name

Command Qualifiers	Defaults
/[NO]AGENT	From existing definition
/[NO]DISPLAY_MENU	From existing definition
/[NO]FINAL=(keyword[,...])	From existing definition
/[NO]INITIAL=(keyword[,...])	From existing definition
/LANGUAGE[=language-name]	From existing definition
/MDB=menu-database-file	From existing definition
/MENU[=menu-path-name]	From existing definition
/PRINTFILE $\left[\begin{array}{l} = \text{print-file-spec} \\ \text{spooled-device-name} \end{array} \right]$	From existing definition
/[NO]SKIPMENULANGUAGE	From existing definition

2.8.11 REMOVE Command (UDU>)

Removes a UDU definition from the user authorization file.

REMOVE user-name

2.8.12 REMOVE/PROXY Command (UDU>)

Removes the specified proxy from the ACMS proxy file (ACMSPROXY.DAT).

REMOVE /PROXY remote-node::remote-user

ACMS Management Utilities and Commands

2.8 UDU Commands

2.8.13 RENAME Command (UDU>)

Changes the user names and, with qualifiers, other information in UDU definitions.

RENAME old-user-name new-user-name

Command Qualifiers	Defaults
/[NO]AGENT	From old definition
/[NO]DISPLAY_MENU	From old definition
/[NO]FINAL=(keyword[,...])	From old definition
/[NO]INITIAL=(keyword[,...])	From old definition
/LANGUAGE[=language-name]	From old definition
/MDB=menu-database-file	From old definition
/MENU[=menu-path-name]	From old definition
/PRINTFILE [= print-file-spec spooled-device-name]	From old definition
/[NO]SKIPMENULANGUAGE	From old definition

2.8.14 SHOW Command (UDU>)

Displays UDU definitions.

SHOW user-name

Command Qualifiers	Defaults
/BRIEF	Full definition

2.8.15 SHOW/PROXY Command (UDU>)

Displays one or more proxies in the ACMS proxy file (ACMSPROXY.DAT).

SHOW /PROXY remote-node::remote-user

ACMS Application Programming Services and Task Debugger Commands

This chapter contains reference material for the programming services supplied by ACMS, and for the ACMS Task Debugger commands.

3.1 ACMS Application Programming Services

This section provides reference material for the ACMS application programming services, `ACMS$GET_TID`, `ACMS$RAISE_NONREC_EXCEPTION`, `ACMS$RAISE_STEP_EXCEPTION`, `ACMS$RAISE_TRANS_EXCEPTION`, and `ACMSAD$REQ_CANCEL`, and the ACMS queuing services, `ACMS$DEQUEUE_TASK` and `ACMS$QUEUE_TASK`.

3.1.1 `ACMS$GET_TID`

Is used by a server procedure to obtain the transaction ID (TID) currently associated with an executing task.

`ACMS$GET_TID (tid.wo.r)`

3.1.2 `ACMS$RAISE_NONREC_EXCEPTION`

Used by a server procedure to raise a nonrecoverable exception and cancel the task.

`ACMS$RAISE_NONREC_EXCEPTION ([exception_code.rl.r])`

3.1.3 `ACMS$RAISE_STEP_EXCEPTION`

Raises a step exception if a step procedure detects an error from which it cannot recover, but which the task definition is able to handle.

`ACMS$RAISE_STEP_EXCEPTION ([exception_code.rl.r])`

3.1.4 `ACMS$RAISE_TRANS_EXCEPTION`

Raises a transaction exception if a step procedure detects an error from which it cannot recover, but which the task definition is able to handle.

`ACMS$RAISE_TRANS_EXCEPTION ([exception_code.rl.r])`

ACMS Application Programming Services and Task Debugger Commands

3.1 ACMS Application Programming Services

3.1.5 ACMSAD\$REQ_CANCEL

Cancels a task. ACMS writes the task cancellation to the Audit Trail Log. When you include a reason parameter, the call also writes the reason for the cancel to the Audit Trail Log.

This service is considered to be a declining feature. It is recommended that you use ACMS\$RAISE_NONREC_EXCEPTION instead.

```
ACMSAD$REQ_CANCEL ((reason.rl.r))
```

3.1.6 ACMS\$DEQUEUE_TASK

Removes or reads a queued task element from the queued task repository and returns information about the task.

```
ACMS$DEQUEUE_TASK (queue_name.rt.dx,  
[element_id.rr.r],  
[flags.rlu.r],  
[ret_task.wt.dx],  
[ret_application.wt.dx],  
[ret_workspace_list.wz.r],  
[ret_workspace_count.wl.r],  
[ret_element_priority.wl.r],  
[ret_username.wt.dx],  
[ret_element_id.wr.r],  
[ret_error_count.wlu.r],  
[ret_last_error.wlu.r],  
[ret_last_error_adt.wadt.r])
```

3.1.7 ACMS\$QUEUE_TASK

Stores the queued task element in an on-disk queued task repository.

```
ACMS$QUEUE_TASK (queue_name.rt.dx,  
task.rt.dx,  
application.rt.dx,  
[workspace_list.rz.r],  
[flags.rlu.r],  
[element_priority.rl.r],  
[username.rt.dx],  
[element_id.wr.r])
```

3.2 ACMS Task Debugger Commands

This section lists the commands available with the ACMS Task Debugger. Use these commands to run an ACMS task without starting an application, to control the task, and to examine and change the contents of the workspaces the task uses as it runs.

3.2.1 @ (At sign) Command

Runs the ACMS Task Debugger commands contained in the named file. The file can contain any ACMS Task Debugger command, including another @ command.

@ file-spec

3.2.2 ACCEPT Command

The ACCEPT command allows the Task Debugger to accept calls from an agent program.

ACCEPT [/qualifier]

Command Qualifiers	Defaults
/CONTINUOUS	None

3.2.3 ASSIGN Command

Assigns a process logical name for a server.

ASSIGN [/qualifier] equivalence-name logical-name

Command Qualifiers	Defaults
/SERVER=server-name	/SERVER=current-server

3.2.4 CANCEL BREAK Command

Removes one or more breakpoints from a task or from all tasks.

CANCEL BREAK [/qualifiers] [breakpoint]

Command Qualifiers	Defaults
/ALL[=task-name]	None

3.2.5 CANCEL TASK Command

Cancels the current task.

CANCEL TASK

ACMS Application Programming Services and Task Debugger Commands

3.2 ACMS Task Debugger Commands

3.2.6 CANCEL TRANSACTION_TIMEOUT Command

Cancels any transaction timeout period previously set.

CANCEL TRANSACTION_TIMEOUT

3.2.7 DEPOSIT Command

Puts a value into a workspace field.

DEPOSIT [/qualifiers] workspace-field-name=value

Command Qualifiers	Defaults
Same as for OpenVMS Debugger	

3.2.8 EXAMINE Command

Displays the contents of a workspace field.

EXAMINE [/qualifiers] workspace-field-name [OF workspace-record-name]

Command Qualifiers	Defaults
Same as for OpenVMS Debugger	

3.2.9 EXIT Command

Ends the debugging session or ends the execution of commands in a command procedure. If typed after the ACMSDBG> prompt, the EXIT command stops all subprocesses started by the Task Debugger and returns to DCL command level. If included in a command procedure, the EXIT command returns control to the command stream that started the command procedure.

EXIT

3.2.10 GO Command

Continues a task after a breakpoint. Also returns to a server process that you left from with `Ctrl/G` and continues any command after an INTERRUPT command.

GO

3.2.11 HELP Command

Displays information about Task Debugger commands, step points, control characters, and symbols.

HELP [topic] [...]

ACMS Application Programming Services and Task Debugger Commands

3.2 ACMS Task Debugger Commands

3.2.12 INTERRUPT Command

Interrupts a server and gives control to the OpenVMS Debugger in that server process. Use this command to get to the DBG> prompt so you can set breakpoints, examine addresses, or change values in a server that has already been started.

```
INTERRUPT server-name [/qualifiers]
```

Command Qualifier	Default
/[TASK=task-name]	None

3.2.13 SELECT Command

Selects and starts a task.

```
SELECT task-name [selection-string]
```

3.2.14 SET BREAK Command

Sets a breakpoint in the task.

```
SET BREAK task-name \ step-name \ location
```

```
SET BREAK task-name \ event
```

3.2.15 SET SERVER Command

Names the server used as the default for the ASSIGN command.

```
SET SERVER server-name
```

3.2.16 SET TRANSACTION_TIMEOUT

Sets the current transaction timeout period.

```
SET TRANSACTION_TIMEOUT seconds
```

3.2.17 SHOW BREAK Command

Displays task-level breakpoints you have set.

```
SHOW BREAK
```

3.2.18 SHOW SERVERS Command

Displays all servers you have started (and not stopped) in the current Task Debugger session.

```
SHOW SERVERS
```

ACMS Application Programming Services and Task Debugger Commands

3.2 ACMS Task Debugger Commands

3.2.19 SHOW TRANSACTION_TIMEOUT

Displays the value of the current transaction timeout.

```
SHOW TRANSACTION_TIMEOUT
```

3.2.20 SHOW VERSION Command

Displays the version number of the Task Debugger.

```
SHOW VERSION
```

3.2.21 START Command

Starts one or more reusable servers.

```
START [/qualifier] [server-name] [...]
```

Command Qualifier	Default
/ALL	None

3.2.22 STEP Command

Runs the task from the current step point to the next task-level step point. When stepping through a task that was called by another task, the Task Debugger proceeds through all the steps in the called task until the task completes.

Control then returns to the parent task where you can continue typing the STEP command to cause the Task Debugger to step through the parent task.

```
STEP
```

3.2.23 STOP Command

Stops one or more servers.

```
STOP [/qualifier] [server-name][,...]
```

Command Qualifier	Default
/ALL	None

Systems Interface (SI) Services

This chapter provides reference material for calling the SI initialization, exchange I/O, submitter, and stream services in agent programs.

4.1 Initialization and Exchange I/O Services

This section provides reference material for calling the SI initialization and exchange I/O services in agent programs.

4.1.1 ACMS\$INIT_EXCHANGE_IO

When the agent is prepared to perform any necessary exchange I/O, ACMS\$INIT_EXCHANGE_IO specifies the type of I/O to perform. It returns an exchange I/O ID. For tasks that use HP DECforms, call ACMS\$INIT_EXCHANGE_IO to open a HP DECforms session.

ACMS\$INIT_EXCHANGE_IO (submitter_id.rq.r,
exchange_io_id.wq.r,
[io_enable_flag.rl.r],
[item_list.rx.r],
[io_capabilities_flag.wl.r])

ACMS\$INIT_EXCHANGE_IO_A (submitter_id.rq.r,
exchange_io_id.wq.r,
[io_enable_flag.rl.r],
[item_list.rx.r],
[io_capabilities_flag.wl.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.1.2 ACMS\$SIGN_IN

Lets an agent sign in a task submitter to ACMS.

ACMS\$SIGN_IN (submitter_id.wq.r,
[username.rt.dx],
[device.rt.dx],
[cancel_routine.rem.r],
[cancel_param.rz.v])

ACMS\$SIGN_IN_A (submitter_id.wq.r,
[username.rt.dx],
[device.rt.dx],
[cancel_routine.rem.r],

Systems Interface (SI) Services

4.1 Initialization and Exchange I/O Services

[cancel_param.rz.v],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.1.3 ACMS\$SIGN_OUT

Lets an agent remove a task submitter from ACMS.

ACMS\$SIGN_OUT (submitter_id.rq.r,
[cancel_flag.rlu.r])

ACMS\$SIGN_OUT_A (submitter_id.rq.r,
[cancel_flag.rlu.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.1.4 ACMS\$TERM_EXCHANGE_IO

This service allows an agent to finish using any I/O initialized during the ACMS\$INIT_EXCHANGE_IO services. ACMS\$TERM_EXCHANGE_IO frees any resources being used by the submitter, for example, HP DECforms sessions or TDMS channels. Any active call on the channel is canceled.

ACMS\$TERM_EXCHANGE_IO (exchange_io_id.rq.r)

ACMS\$TERM_EXCHANGE_IO_A (exchange_io_id.rq.r,
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.2 Submitter Services

This section provides reference material for the submitter services.

4.2.1 ACMS\$CALL

Submits an ACMS task. This service completes when the task ends. If you use the asynchronous ACMS\$CALL_A service, you also must call the ACMS\$WAIT service.

ACMS\$CALL ([submitter_id.rq.r],
procedure_id.rq.r,
arguments.rz.r,
[tid.ro.r])

ACMS\$CALL_A ([submitter_id.rq.r],
procedure_id.rq.r,
arguments.rz.r,
[tid.ro.r],
[comp_status.wq.r],
[efn.rbu.r],

[astadr.szem.r],
[astprm.rz.v])

4.2.2 ACMS\$CANCEL_CALL

Cancels a task started by the task submitting agent. This service only cancels tasks started with ACMS\$START_CALL. The agent must also use the ACMS\$WAIT_FOR_CALL_END service with this service to get notification of the call canceling.

ACMS\$CANCEL_CALL ([submitter_id.rq.r],
call_id.rq.r,
[reason_code.rlu.r])

ACMS\$CANCEL_CALL_A ([submitter_id.rq.r],
call_id.rq.r,
[reason_code.rlu.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.2.3 ACMS\$GET_PROCEDURE_INFO

Finds and returns the I/O method (terminal, request, stream, or none), the procedure ID for the task, and the number of workspace arguments the agent can pass to a task in an ACMS application.

ACMS\$GET_PROCEDURE_INFO ([submitter_id.rq.r],
procedure.rt.dx,
package.rt.dx,
item_list.rx.r)

ACMS\$GET_PROCEDURE_INFO_A ([submitter_id.rq.r],
procedure.rt.dx,
package.rt.dx,
item_list.rx.r,
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.2.4 ACMS\$START_CALL

Submits an ACMS task. This service completes when the task has been submitted. It returns a call ID to the agent.

ACMS\$START_CALL ([submitter_id.rq.r],
procedure_id.rq.r,
call_id.wq.r,
arguments.rz.r,
[tid.ro.r])

ACMS\$START_CALL_A ([submitter_id.rq.r],
procedure_id.rq.r,
call_id.wq.r,

Systems Interface (SI) Services

4.2 Submitter Services

arguments.rz.r,
[tid.ro.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.2.5 ACMS\$WAIT_FOR_CALL_END

Waits for a task to complete. This service only waits for tasks started with ACMS\$START_CALL. This service also reports access errors that occurred after the task was submitted.

ACMS\$WAIT_FOR_CALL_END (submitter_id.rq.r,
call_id.rq.r)

ACMS\$WAIT_FOR_CALL_END_A (submitter_id.rq.r,
call_id.rq.r,
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.3 Stream Services

This section provides reference material for calling the SI stream services in agent programs.

4.3.1 ACMS\$REPLY_TO_STREAM_IO

Reacts to I/O requests on the stream. The agent must gather information for the ACMS\$WAIT_FOR_STREAM_IO input string and fill the string before calling this service.

ACMS\$REPLY_TO_STREAM_IO (connect_id.rq.r,
io_id.wq.r,
[io_status.rl.r])

ACMS\$REPLY_TO_STREAM_IO_A (connect_id.rq.r,
io_id.wq.r,
[io_status.rl.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.3.2 ACMS\$WAIT_FOR_STREAM_IO

Waits for I/O messages. This service completes when the application execution controller (EXC) executes a READ or WRITE clause in the task definition.

ACMS\$WAIT_FOR_STREAM_IO (connect_id.rq.r,
output_object.wz.r,
input_object.wz.r,
io_id.wq.r,

	[cancel_routine.rem.r], [cancel_param.rz.v])
ACMS\$WAIT_FOR_STREAM_IO_A	(connect_id.rq.r, output_object.wz.r, input_object.wz.r, io_id.wq.r, [comp_status.wq.r], [efn.rbu.r], [astadr.szem.r], [astprm.rz.v], [cancel_routine.rem.r], [cancel_param.rz.v])

4.4 Superseded Services

This section provides reference material for six services used in earlier versions of ACMS.

4.4.1 ACMS\$CLOSE_RR

Closes a TDMS channel to a terminal and disassociates it from a submitter ID. Any active TDMS call on the channel is canceled.

ACMS\$CLOSE_RR	([channel.rlu.r], [nullarg])
ACMS\$CLOSE_RR_A	([channel.rlu.r], [nullarg], [comp_status.wq.r], [efn.rbu.r], [astadr.szem.r], [astprm.rz.v])

4.4.2 ACMS\$CONNECT_STREAM

Establishes a connection to a stream and returns a connect ID. Before using this service, you have to create a stream with ACMS\$CREATE_STREAM.

ACMS\$CONNECT_STREAM	(stream.id.rq.r, mode.rl.r, connect.id.wq.r, [submitter.id.rq.r])
ACMS\$CONNECT_STREAM_A	(stream.id.rq.r, mode.rl.r, connect.id.wq.r, [comp_status.wq.r], [efn.rbu.r], [astadr.szem.r], [astprm.rz.v], [submitter_id.rq.r])

Systems Interface (SI) Services

4.4 Superseded Services

4.4.3 ACMS\$CREATE_STREAM

Creates a stream and returns the stream identification.

ACMS\$CREATE_STREAM (mode.rl.r,
stream_id.wq.r)

ACMS\$CREATE_STREAM_A (mode.rl.r,
stream_id.wq.r,
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.4.4 ACMS\$DELETE_STREAM

Deletes a stream. Use this service after ACMS\$DISCONNECT_STREAM disconnects all connect IDs to the stream. Once deleted, a stream is not available for use by other tasks.

ACMS\$DELETE_STREAM (stream_id.rq.r,
[flags.rl.r])

ACMS\$DELETE_STREAM_A (stream_id.rq.r,
[flags.rl.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.4.5 ACMS\$DISCONNECT_STREAM

Breaks a connection to a stream. The application execution controller (EXC) must disconnect from the stream before the agent can disconnect.

ACMS\$DISCONNECT_STREAM (connect_id.rq.r,
[flags.rl.r])

ACMS\$DISCONNECT_STREAM_A (connect_id.rq.r,
[flags.rl.r],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

4.4.6 ACMS\$OPEN_RR

For tasks that use TDMS, the agent calls ACMS\$OPEN_RR to open a TDMS channel to a terminal and associates it with a submitter ID. Subsequent task selections for that submitter use the channel for all task request I/O, including remote request I/O. For tasks that use the ACMS Request Interface (RI), the agent calls ACMS\$OPEN_RR to prepare the agent process to do the I/O.

Systems Interface (SI) Services 4.4 Superseded Services

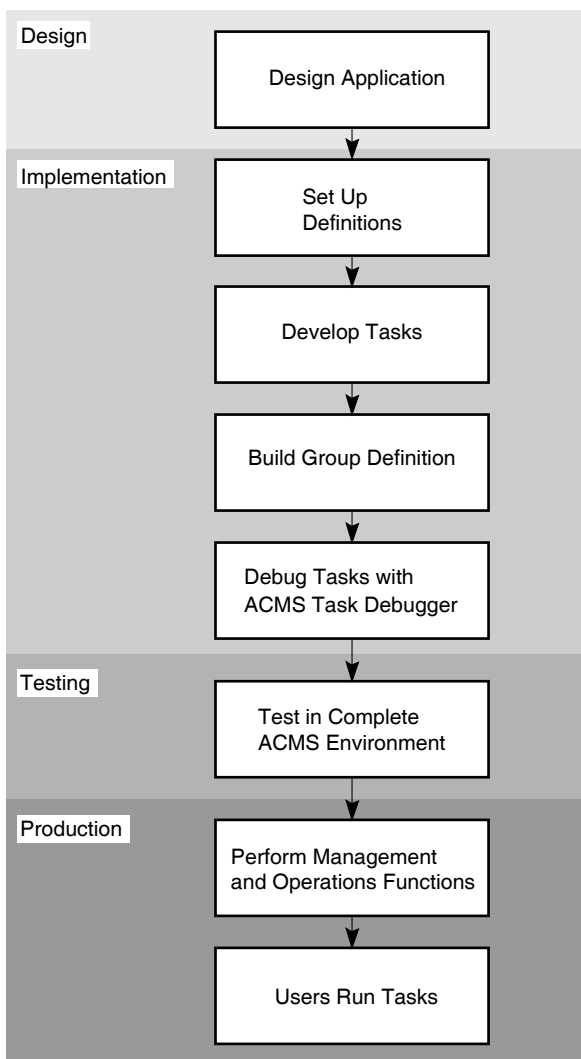
ACMS\$OPEN_RR (device.rt.dx,
channel.wlu.r,
[submitter_id.rq.r],
[flags.rl.r],
[nullarg])

ACMS\$OPEN_RR_A (device.rt.dx,
channel.wlu.r,
[submitter_id.rq.r],
[flags.rl.r],
[nullarg],
[comp_status.wq.r],
[efn.rbu.r],
[astadr.szem.r],
[astprm.rz.v])

Checklist for ACMS Application Development

Figure A-1 illustrates the phases of application development for ACMS, from the initial design of an application to the actual production of the application. Figure A-1 is followed by a more detailed checklist of the phases of application development you can use with the figure.

Figure A-1 ACMS Application Design, Development, and Use



TAY-0094-AD

Checklist for ACMS Application Development

1. Design application
 - Analyze business
 - Design application
 - Create preliminary definition and other specifications
2. Set up data
 - RMS:
 - Define the records (CDO)
 - Create the files (RMS utilities or DCL commands)
 - DBMS:
 - Define the schema, subschema, storage schema (DDL)
 - Create the database (DBO)
 - Rdb:
 - Define the records and their relationships (SQL or RDO)
 - Create the database (SQL or RDO)
3. Define tasks
 - Create task definitions (ADU)
4. Create workspaces
 - Create a file containing record definition (editor)
 - Store the definition in the data dictionary (CDO)
5. Create HP DECforms forms
 - Create the panel using the panel editor (FDE)
 - Edit the IFDL file
 - Translate the IFDL file to the FORM file (IFDL translator)
6. Create TDMS forms
 - Define background text, input and output fields (FDU)
 - Define the help form (FDU)
7. Create TDMS requests
 - Define a request (RDU)
 - Define a request library (RDU)
 - Build the request library (RDU)
8. Write procedures
 - Write and compile step procedures
 - Write and compile initialization and termination procedures for each procedure server
 - Write and compile cancel procedures
9. Create message files
 - Create message source files (editor or DCL)

Checklist for ACMS Application Development

- Generate object modules (OpenVMS Message Utility)
 - Generate executable message files (DCL LINK command)
10. Create task groups
 - Define the task group including tasks and servers (ADU)
 - Build the task group using the /DEBUG qualifier (ADU)
 11. Create the procedure server image
 - Link the task group transfer module, the message file object module, and the object modules for step procedures, cancel procedures, and initialization and termination procedures (LINK/DEBUG command)
 12. Debug
 - Run tasks under ACMS Task Debugger and OpenVMS Symbolic Debugger
 - Revise definitions and programs and test again
 13. Set up the application
 - Create the application definition (ADU)
 - Build the application database (ADU)
 14. Move the application database to ACMS\$DIRECTORY
 - Use the DCL COPY command
 - Or, create the application authorization definition (AAU) and use the ACMS/INSTALL command
 15. Set up menus
 - Create menu definitions (ADU)
 - Build the menu database (ADU)
 16. Test the application
 - Start the application and run tasks to test implementation
 17. Authorize users
 - Authorize users for access to OpenVMS (OpenVMS Authorize Utility)
 - Authorize users for access to ACMS, assign the menu database and initial the menu (UDU)
 18. Authorize terminals
 - Authorize terminals for access to ACMS (DDU)
 - Define whether terminals are controlled by ACMS or OpenVMS (DDU)
 19. Manage and tune the application
 - Start and stop application
 - Monitor and tune application
 - Modify application, servers, tasks, and menus as necessary

Changing and Debugging ACMS Applications

Application development is a cyclical process. Omissions or problems in the analysis or design might not show up until implementation is complete. To correct these omissions or problems, you might need to redo part of the design and implementation. For this reason, it is helpful to understand what parts of an application have to change if you change some other part.

Table B-1 summarizes these relationships between different parts of ACMS applications.

Table B-1 Changing ACMS Applications

Changed Component	Changes to Related Components	When Change Takes Effect
Menu database name (.MDB)	User definitions (in ACMSUDF.DAT file) pointing to menu database must be changed.	Next time user signs in to ACMS.
	Menu definitions should be changed if they include a clause naming database file.	When user signs in after menu database is rebuilt.
Menu definition	Menu database containing that definition must be rebuilt. New database must be put in directory pointed to by ACMSUDF.DAT records (usually ACMS\$DIRECTORY).	When user signs in after menu database is rebuilt.
Application database name (.ADB)	Menu definitions pointing to tasks in that application must be changed and menu database rebuilt.	When user signs in after menu database is rebuilt.
	Application authorization definitions must be modified to reflect new application name.	When application database is reinstalled with ACMS/INSTALL command.
Application definition	Application database must be rebuilt. Menu definition database does not need to be changed unless name of task is changed in application definition.	When application is stopped and restarted.
	If changes to application definition conflict with existing authorization for application in ACMSAAF.DAT, authorization must be modified and application database reinstalled with ACMS/INSTALL command.	When application is moved to ACMS\$DIRECTORY with ACMS/INSTALL command.

(continued on next page)

Changing and Debugging ACMS Applications

Table B-1 (Cont.) Changing ACMS Applications

Changed Component	Changes to Related Components	When Change Takes Effect
Task group database name (.TDB)	Application definition must be changed and rebuilt. Task group definition should be changed and rebuilt if it points to task group database file.	When application is stopped and restarted.
Task group definition	<p>Task group database must be rebuilt.</p> <p>If a task was added or removed, the application database must be rebuilt, even if no change is required in application definition.</p> <p>If a task is removed or a task name changed, menu definition pointing to task must be changed. In addition, the menu database and application database must be rebuilt. If the task removed from the task group was also named in application definition, application definition must be changed and rebuilt.</p> <p>If default control attributes of a task are changed, and these attributes are not overridden in application definition, application database must be rebuilt.</p> <p>If changes to task group definition conflict with existing authorization for application in ACMSAAF.DAT, authorization must be modified and application database reinstalled with ACMS/INSTALL command.</p>	<p>When application is stopped and restarted.</p> <p>When application is moved to ACMS\$DIRECTORY with ACMS/INSTALL command.</p>
Task definition	<p>Task group database must be rebuilt.</p> <p>If default control attributes of task are changed, and these attributes are not overridden in application definition, application database must be rebuilt.</p> <p>If task name is changed, task group definition must be changed to reflect new name.</p>	When application is stopped and restarted.
Request	<p>Request library (.RLB) must be rebuilt.</p> <p>If number of records or if record definitions used by request are changed, task group database must be rebuilt.</p> <p>If request name is changed, task definition must be changed.</p> <p>If request library name is changed, task group definition must be changed.</p>	When application is stopped and restarted.

(continued on next page)

Changing and Debugging ACMS Applications

Table B–1 (Cont.) Changing ACMS Applications

Changed Component	Changes to Related Components	When Change Takes Effect
Form definition	Request library (.RLB) must be rebuilt. If form name is changed, request definition must be changed.	When application is stopped and restarted.
Form panel	FORM file must be back-translated to produce a source IFDL file. Object module must be extracted and relinked with or without escape units.	When application is stopped and restarted.
Form record	IFDL file must be translated to produce a new FORM file. Object module must be extracted and relinked with or without escape units.	When application is stopped and restarted.
Step procedure	Step procedure must be recompiled and server image (.EXE) relinked If number of workspaces or if record definitions that the procedure uses for workspaces are changed, task must be redefined and task group database rebuilt. If procedure name is changed, procedure server definition in task group must be changed. Task definition must be changed.	When application is stopped and restarted, or when server is replaced.
Message source file	File of message texts (.EXE) must be relinked. If message was added or removed or order of messages changed, new message object module must be generated. Server image must be relinked with the new module.	When server using message is restarted after relink. Stop and restart application to ensure that changes are available.
Workspace definition	If task definition, form record, request, or procedure refers to fields that have changed name, definition or program must be changed. Task group database or request library must be rebuilt, or procedure relinked. Both task group and form file or request library must also be rebuilt and program relinked if order of fields in workspace or other characteristics have changed, even if requests, task definitions, and procedures are not affected by change.	When application is stopped and restarted after rebuild.

(continued on next page)

Changing and Debugging ACMS Applications

Table B–1 (Cont.) Changing ACMS Applications

Changed Component	Changes to Related Components	When Change Takes Effect
Application (with AAU)	If name of workspace definition is changed, procedure, task definition, and form record or request must also be changed. Task group and request library must be rebuilt. Procedure must be recompiled and relinked. If using ACMS/INSTALL command, reinstall application database.	When application database is moved into ACMS\$DIRECTORY with ACMS/INSTALL command.
User or device authorization (with UDU or DDU)	If changes to authorization place further restrictions on application, remove old application from ACMS\$DIRECTORY and reinstall application database. If change does not affect access control lists or menu names, no other change is required. If change affects access control lists in application definition, application must be redefined and rebuilt. If change affects names of menus, menus must be redefined and rebuilt. If change affects which terminals are controlled by ACMS, no other change is required.	Next time user signs in to ACMS. After ACMS/RESET TERMINALS command is issued, or after ACMS is stopped and restarted.

Table B–2 describes the files needed to run a task with the ACMS Task Debugger.

Table B–2 Files Used to Debug ACMS Tasks

File	Description
Data files or database files for task group	Created and populated using either RMS, DBMS, or Rdb.
Message file or files for task group	Created with the OpenVMS Message Utility. This file contains text of messages and their message symbols. You need this file only if your tasks use the GET MESSAGE clause to access a message file that is separate from the server image.
Procedure Server Images	Created with the LINK command. Contains executable versions of the procedure server transfer module, message file module, and all procedures for the task group.
HP DECforms form files	Created using HP DECforms.

(continued on next page)

Changing and Debugging ACMS Applications

Table B–2 (Cont.) Files Used to Debug ACMS Tasks

File	Description
Task database — TDB	Created with the BUILD command of the ACMS Application Definition Utility — ADU. Contains information used by ACMS to run tasks. Include the /DEBUG qualifier with the BUILD command to examine and deposit data in the workspace while debugging the task.

Table B–3 describes the sources used to create the files used for debugging ACMS tasks.

Table B–3 Source Files for Debugging

File	Description
ADU input files	Contain source definitions for task groups and tasks.
CDO input files	Contain CDD source definitions for ACMS workspace records.
RDU input files	Contain source definitions for requests and request libraries.
Step, initialization, termination, and cancel procedures	Written in COBOL, BASIC, or other high-level languages.
Message source files	Contain source text for messages.
Procedure server transfer module	Created by building the task group. Contains vectors for the procedures handled by the server and the main entry point for the procedure server image.

Summary of ACMS System Workspaces

Each of the three ACMS system workspaces has a different purpose. All of the Common Data Definition Language (CDDL) record definitions for these workspaces are stored in the CDD\$TOP.ACMS\$DIR.ACMS\$WORKSPACES directory in the CDD. This appendix lists these workspaces and explains the uses of each.

C.1 ACMS\$PROCESSING_STATUS System Workspace

The ACMS\$PROCESSING_STATUS workspace handles processing status information. It has four fields, each for a different part of that information. The CDD location of the CDDL record definition for this workspace is CDD\$TOP.ACMS\$DIR.ACMS\$WORKSPACES.ACMS\$PROCESSING_STATUS. Table C-1 describes the fields in the ACMS\$PROCESSING_STATUS workspace.

Table C-1 Fields in ACMS\$PROCESSING_STATUS

ACMS\$PROCESSING_STATUS Workspace	
ACMS\$L_STATUS	
Type:	Signed longword
Description:	Contains the return status from the last processing step. The initial value of the ACMS\$L_STATUS field is set to 1 (SUCCESS) when a task is started.
ACMS\$T_SEVERITY_LEVEL	
Type:	Text
Size:	1 character
Description:	Contains a single-character severity level code representing the return status in the ACMS\$L_STATUS field. The characters this field can contain are: S (SUCCESS), I (INFORMATION), W (WARNING), E (ERROR), F (FATAL), ? (OTHER). The initial value of ACMS\$T_SEVERITY_LEVEL is "S".
ACMS\$T_STATUS_TYPE	
Type:	Text
Size:	1 character
Description:	Contains a single character indicating the severity level of the return status in the ACMS\$L_STATUS field. A "G" indicates the low bit in the ACMS\$L_STATUS field is set to 1. A "B" indicates the low bit is clear. The initial value of the ACMS\$T_STATUS_TYPE field is "G".

(continued on next page)

Summary of ACMS System Workspaces

C.1 ACMS\$PROCESSING_STATUS System Workspace

Table C–1 (Cont.) Fields in ACMS\$PROCESSING_STATUS

ACMS\$PROCESSING_STATUS Workspace	
ACMS\$T_STATUS_MESSAGE/ACMS\$T_STATUS_MESSAGE_LONG	
Type:	Text
Size:	80/132 characters
Description:	ACMS\$T_STATUS_MESSAGE is an 80-character variant of the 132-character ACMS\$T_STATUS_MESSAGE_LONG field. When you use the GET ERROR MESSAGE clause, this field contains the error message associated with the return status code in ACMS\$L_STATUS. The ACMS\$T_STATUS_MESSAGE_LONG field is set initially to spaces.

C.2 ACMS\$SELECTION_STRING System Workspace

The ACMS\$SELECTION_STRING workspace handles strings passed by a task submitter (terminal user) at task selection time. It has a single field. The CDD location of the CDDL record definition for this workspace is CDD\$TOP.ACMS\$DIR.ACMS\$WORKSPACES.ACMS\$SELECTION_STRING. Table C–2 describes the field in the ACMS\$SELECTION_STRING workspace.

Table C–2 Fields in ACMS\$SELECTION_STRING

ACMS\$SELECTION_STRING Workspace	
ACMS\$T_SELECTION_STRING	
Type:	Text
Size:	255 characters
Description:	Contains the selection string provided by a terminal user at task selection time. If the user does not provide a selection string, ACMS sets the field to spaces. If the task is a queued task, the first 32 bytes of the selection string contain the queued task element ID.

C.3 ACMS\$TASK_INFORMATION System Workspace

The ACMS\$TASK_INFORMATION workspace handles task execution information. It has 10 fields, each for a different part of that information. The CDD location of the CDDL record definition for this workspace is CDD\$TOP.ACMS\$DIR.ACMS\$WORKSPACES.ACMS\$TASK_INFORMATION. Table C–3 describes the fields in the ACMS\$TASK_INFORMATION workspace.

Table C–3 Fields in ACMS\$TASK_INFORMATION

ACMS\$TASK_INFORMATION Workspace	
ACMS\$AL_TASK_ID	
Type:	Signed longword array
Size:	4 longwords

(continued on next page)

Summary of ACMS System Workspaces

C.3 ACMS\$TASK_INFORMATION System Workspace

Table C-3 (Cont.) Fields in ACMS\$TASK_INFORMATION

ACMS\$TASK_INFORMATION Workspace	
ACMS\$AL_TASK_ID	
Description:	<p>Contains the task ID in binary format for the current task instance; the ACMS\$AL_TASK_ID field is a four-element longword array.</p> <p>It is possible that two task instances can have the same value, if the tasks have been selected on two different nodes. To ensure a unique task identifier, use both the ACMS\$AL_TASK_ID field and the ACMS\$T_SUBMITTER_NODE field.</p>
ACMS\$L_TASK_SEQUENCE_NUMBER	
Type:	Signed longword
Description:	<p>Contains the number of the current task instance within the current task; the content of this field is always one (1) when the task is initially selected from a menu. ACMS increments this number each time the user repeats the task or chains to another task, thus starting a new task instance without returning to the menu.</p>
ACMS\$T_TASK_NAME	
TYPE:	Text
Size:	31 characters
Description:	<p>Contains the task name as defined in the application under which the task is running. ACMS does not update this field when a task chains to another task.</p>
ACMS\$T_TASK_IO_DEVICE	
Type:	Text
Size:	8 characters
Description:	<p>Contains the device name for the task submitter. For remote users, the device name is always NL. For local request I/O or terminal I/O users, this field includes the terminal device name. For stream I/O or no I/O, this field is set to spaces.</p> <p>If this field contains a device name (not spaces or NL), then the device can be used by the task to perform I/O from a processing step.</p>
ACMS\$AL_TASK_SUBMITTER_ID	
Type:	Signed longword array
Size:	4 longwords
Description:	<p>Contains the current terminal user's identification code for the user who started the current task instance. This field is a four-element longword array.</p>

(continued on next page)

Summary of ACMS System Workspaces

C.3 ACMS\$TASK_INFORMATION System Workspace

Table C-3 (Cont.) Fields in ACMS\$TASK_INFORMATION

ACMS\$TASK_INFORMATION Workspace	
ACMS\$T_TASK_USERNAME	
Type:	Text
Size:	12 characters
Description:	Contains the OpenVMS user name for the terminal user who started the current task instance. For remote tasks, this is the name of the proxy.
ACMS\$T_SUBMITTER_NODE_NAME	
Type:	Text
Size:	15 characters
Description:	Contains the DECnet node name for the task submitter.
ACMS\$L_CALL_SEQUENCE_NUMBER	
Type:	Signed longword
Description:	Contains the call sequence number of the currently called task. ACMS increments this number each time a task calls another task.
ACMS\$T_SIGN_IN_USERNAME	
Type:	Text
Size:	12 characters
Description:	<p>Contains the OpenVMS user name of the user on the submitter node.</p> <p>If a submitter selects a remote task, then the user name under which that task runs may be different from the user name under which the task is signed in. The contents of the ACMS\$T_TASK_USERNAME is based on the proxy lookup and user name defaulting mechanism and may differ from the ACMS\$T_SIGN_IN_USERNAME field.</p> <p>If a submitter selects a local task, the ACMS\$T_SIGN_IN_USERNAME field will be the same as the ACMS\$T_TASK_USERNAME field.</p> <p>To distinguish between users that have the same name but reside on different nodes, use the ACMS\$T_SIGN_IN_USERNAME field with the ACMS\$T_SUBMITTER_NODE_NAME field to log the user name and the node location.</p>

(continued on next page)

Summary of ACMS System Workspaces

C.3 ACMS\$TASK_INFORMATION System Workspace

Table C-3 (Cont.) Fields in ACMS\$TASK_INFORMATION

ACMS\$TASK_INFORMATION Workspace	
ACMS\$T_SIGN_IN_DEVICE	
Type:	Text
Size:	8 characters
Description:	<p>Contains the name of the device that was supplied to ACMS when the submitter signed in.</p> <p>For applications using the ACMS command process, this field contains a terminal device name.</p> <p>For applications using a user-written command process (agent), this field can contain a terminal device name, the name of a nonterminal device that the agent is handling, or the NL device specification.</p> <p>Use the ACMS\$T_SIGN_IN_DEVICE field with the ACMS\$T_SUBMITTER_NODE_NAME field to log the device name and its node location. It is necessary to use both of these fields if you wish to distinguish between devices that have the same name but are residing on different nodes.</p>
