

SysWorks V3.5

Release Notes

Order Number SWRK-RN-351

August 2009

This manual briefly describes the new and modified features provided in SysWorks V3.5.

- Revision/Update Information:** This manual supercedes the SysWorks V3.4-1 Release Notes
- Operating System and Version:** OpenVMS VAX V7.2 or higher;
OpenVMS Alpha V7.2 or higher;
DECwindows/Motif V1.2-3 or higher
- Software Version:** SysWorks V3.5

© Corpita Pty Ltd 1987 - 2009.

All Rights Reserved.

Printed in Australia

The following are trademarks of Compaq Computer Corporation: ACMS, ALL-IN-1, AXP, BASIC, Bookreader, CDA, CI, DATATRIEVE, DBMS, DDIF, DEC, DEC ACCESSWORKS, DEC Ada, DEC C, DEC Fortran, DEC Pascal, DECdecision, DECdesign, DECdirect, DECdns, DECdocument, DECdtm, DECforms, DECimage, DECintact, DECmigrate, DECnet, DECnet/OSI, DECset, DECsupport, DECTp, DECwindows, Digital, DTIF, EDT, HSC, MASSBUS, MicroVAX, MicroVAX II, MSCP, OpenVMS, OpenVMS Cluster, RA, StorageWorks, TA, TMSCP, TURBOchannel, ULTRIX, VAX, VAX C, VAX MACRO, VAX-11/780, VAXcluster, VAXELN, VAXft, VAXstation, VIDA, VMS, VMScluster, VT100, and the DIGITAL logo.

PostScript is a registered trademark of Adobe Systems Inc.

Motif is a registered trademark of Open Software Foundation, Inc.

Oracle is a registered trademark, and Oracle CDD/Repository, Oracle CODASYL DBMS, Oracle Expert, Oracle Rally, Oracle Rdb, Oracle Trace and Rdb7 are trademarks of Oracle Corporation.

OSI is a registered trademark of CA Management, Inc

All other trademarks and registered trademarks are the property of their respective holders.

This document was prepared using DECdocument V3.3.

Contents

Send Us Your Comments	vii
Preface	ix
1 Summary of Changes	
1.1 Summary of Changes for V3.5	1-1
1.2 Summary of Changes for V3.4-1	1-1
1.3 Summary of Changes for V3.4	1-1
1.4 Summary of Changes for V3.3	1-1
1.5 Summary of Changes for V3.2	1-2
2 New and Modified Features	
2.1 Documentation Updates	2-1
2.1.1 Traditional Method	2-1
2.1.2 Replicator Method	2-2
2.2 Concepts	2-2
2.2.1 Alternative Installation Directory	2-2
2.2.2 Application Environment Scope	2-3
2.2.3 Error Handling	2-3
2.2.4 HTTP File Types	2-3
2.2.5 Installation Level	2-3
2.2.6 Library Module Deletion	2-4
2.2.7 Message Help Support	2-4
2.2.8 Message Logging	2-4
2.2.9 Multi-architecture Support	2-4
2.2.10 Multi-variant Support	2-5
2.2.11 Multi-version Support	2-6
2.2.12 OpenVMS Alpha 8.3	2-6
2.2.13 OpenVMS Alpha	2-6
2.2.14 Oracle Rdb Version	2-6
2.2.15 Secure Task Actions	2-6
2.2.16 Security Model Changed	2-7
2.2.17 SPECIFIC Scope	2-7
2.2.18 Subprocess Editor	2-7
2.2.19 Synchronized Batch Jobs Status Message	2-8

2.2.20	Meta object Model	2-8
2.2.20.1	V3.4	2-8
2.2.20.2	V3.3	2-8
2.2.20.3	V3.2	2-8
2.2.20.4	V3.1	2-9
2.2.20.5	V3.0	2-9
2.2.21	Test Directory	2-10
2.2.22	Utility Logicals Scope	2-10
2.2.23	VARIANT Context	2-11
2.3	Layered Products	2-11
2.3.1	New License Recognition	2-11
2.3.2	New Product Support	2-11
2.3.3	ACA Services Support	2-12
2.3.4	DECdocument	2-13
2.3.5	DECset V12.1	2-13
2.3.6	DECset V12	2-13
2.3.7	DECset V11	2-13
2.3.8	Java Support	2-13
2.3.9	ObjectBroker Support	2-13
2.4	Commands	2-13
2.4.1	ACCEPT	2-14
2.4.2	BUILD	2-14
2.4.3	BUILD and COMPILE	2-16
2.4.4	BUILD, CONTEXT APPLICATION and CONTEXT ENVIRONMENT	2-16
2.4.5	CHANGE	2-16
2.4.6	CHGCTL	2-16
2.4.7	COMMON	2-16
2.4.8	COMPARE	2-16
2.4.9	CONTEXT	2-16
2.4.10	CVTMMS	2-17
2.4.11	CVTMMS	2-17
2.4.12	CVTRDB	2-17
2.4.13	DEC Rdb Precompiler Commands	2-17
2.4.14	DIRTOOLS	2-17
2.4.15	DO	2-18
2.4.16	DEVTOOLS	2-18
2.4.16.1	ADU	2-19
2.4.16.2	ATTACH	2-19
2.4.16.3	CDO	2-19
2.4.16.4	CMS FETCH	2-19
2.4.16.5	CMS PROMOTE	2-19
2.4.16.6	CMS REPORT	2-19
2.4.16.7	CMS RESERVE	2-19
2.4.16.8	CMS SHOW GENERATION	2-19
2.4.16.9	CONVERT/GENERATE	2-20
2.4.16.10	DELETE	2-20
2.4.16.11	DIFFERENCES/DATES	2-20
2.4.16.12	RDU	2-20
2.4.16.13	SET UTILITY/KEEP	2-20
2.4.16.14	SET UTILITY/NOKEEP	2-20
2.4.16.15	SHOW CONTEXT	2-20
2.4.17	EDIT	2-20
2.4.18	FDU	2-21
2.4.19	LADCP and LASTCP	2-21

2.4.20	LOGOUT	2-21
2.4.21	MAILBOX/DELETE	2-21
2.4.22	REDUCE	2-21
2.4.23	SC	2-21
2.4.24	SPECIFIC	2-21
2.4.25	SRCCTL	2-22
2.4.26	TSTCTL	2-22
2.4.27	UTLTOOLS	2-22
2.4.27.1	DEFINE	2-22
2.4.27.2	General	2-22
2.4.28	VSNCTL	2-22
2.4.28.1	TRIAL	2-23
2.4.29	WHO	2-23
2.5	Menus	2-23
2.5.1	CHGCTL, SRCCTL and TSTCTL Menu Help	2-23
2.5.2	Environment Registration Menu	2-23
2.6	Tasks	2-23
2.6.1	Merge maintenance into development	2-23
2.7	Logical Names	2-23
2.7.1	<i>appl_ACMS_REPLACE</i>	2-23
2.7.2	<i>appl_APPLICATIONS</i>	2-23
2.7.3	<i>appl_CMS_CLASS</i>	2-24
2.7.4	<i>appl_CMS_GENERATION</i>	2-24
2.7.5	<i>appl_CMS_GROUP</i>	2-24
2.7.6	<i>appl_CMS_PATH</i>	2-24
2.7.7	<i>appl_CMS_VARIANT</i>	2-25
2.7.8	<i>envr_DEVELOPER_STYLE</i>	2-25
2.7.9	<i>appl_GROUPS</i>	2-25
2.7.10	<i>appl_MENUS</i>	2-25
2.7.11	<i>appl_MSGHLP_LIB</i>	2-25
2.7.12	<i>appl_TASKS</i>	2-25
2.7.13	<i>appl_TDMS_REPLACE</i>	2-26
2.7.14	SWDEV_DEVELOPER_STYLE	2-26
2.7.15	SWRK_ENVR_ <i>envr</i>	2-26
2.7.16	SWRK_ENVR_VACL_ <i>envr</i>	2-27
2.7.17	SWRK_ENVR_WRK_ <i>envr</i>	2-27
2.7.18	SWRK_UTILITY_LOGICALS_SCOPE	2-27
2.7.19	SQL\$USER and LNK\$LIBRARY_ <i>nn</i>	2-27
2.8	Routines	2-28
2.8.1	SWRK_BINARY_SEARCH	2-28
2.8.2	SWRK_CLOSE_OUTPUT	2-28
2.8.3	SWRK_DISPLAY_HELP	2-28
2.8.4	SWRK_DUMP_IMAGE_INFO	2-28
2.8.5	SWRK_ESTABLISH	2-28
2.8.6	SWRK_EXCEPTION_HANDLER	2-28
2.8.7	SWRK_EXTENDED_FILE_SEARCH	2-29
2.8.8	SWRK_HANDLE_IMAGE_VECTOR	2-29
2.8.9	SWRK_HANDLE_QUALIFIERS	2-29
2.8.10	SWRK_OPEN_OUTPUT	2-29
2.8.11	SWRK_PUT_OUTPUT	2-29
2.9	Macros	2-29
2.9.1	FIXUP	2-29
2.9.2	UPDATE	2-29
2.10	MMS Generators	2-29
2.10.1	Improvements to All MMS Generators	2-29

2.10.2	ACMS	2-30
2.10.3	ACMS Task Group	2-30
2.10.4	ADA	2-30
2.10.5	Basic	2-30
2.10.6	Bookreader	2-31
2.10.7	C	2-31
2.10.8	C	2-31
2.10.9	C++	2-31
2.10.10	Cobol	2-32
2.10.11	DECdocument	2-32
2.10.12	DECdocument Graphics	2-32
2.10.13	DECforms	2-32
2.10.14	DECforms	2-33
2.10.15	Linker	2-33
2.10.16	Macro	2-33
2.10.17	Pascal	2-33
2.10.18	Linker	2-33
2.11	Command Procedures	2-33
2.11.1	SYLOGIN.COM	2-34
2.12	DECwindows	2-34
2.12.1	DECwindows Utilities Pulldown Menu	2-34
2.12.2	SysWorks Manage Pulldown Menu	2-34
2.12.3	CPU Load and Eyes	2-34
2.12.4	DECwindows/Motif	2-34
2.12.5	.DOC and .DOC_STYLE File Types	2-34
2.13	Miscellaneous	2-34
2.13.1	SWDEV_VERY_OLD_FILE.TXT	2-34
2.13.2	Message Symbol Code Prefix Change	2-34
2.13.3	Display of messages in hook files	2-34

3 Problems Fixed

3.1	Commands	3-1
3.1.1	Many Commands	3-1
3.1.2	ACCEPT	3-1
3.1.3	CONTEXT	3-1
3.1.4	DEVTOOLS	3-1
3.1.4.1	CMS CREATE ELEMENT	3-2
3.1.4.2	CMS DELETE ELEMENT/[NO]CONFIRM	3-2
3.1.4.3	CMS FETCH	3-2
3.1.4.4	CMS INSERT GENERATION	3-2
3.1.4.5	CMS REPLACE	3-2
3.1.4.6	CMS RESERVE	3-2
3.1.4.7	CONVERT/GENERATE	3-2
3.1.4.8	DIFFERENCES/DATES	3-3
3.1.4.9	DIFFERENCES/DATES	3-3
3.1.5	CHGCTL	3-3
3.2	Menus	3-3
3.3	Routines	3-3
3.3.1	SWRK_HANDLE_QUALIFIERS	3-3
3.3.2	SWRK_MOVE_FILE	3-3
3.4	Tasks	3-3
3.4.1	Rename a disk volume	3-3
3.4.2	Merge maintenance into development	3-4

3.5	MMS Generators	3-4
3.5.1	ACMS Menu Definition	3-4
3.5.2	CDD/Repository	3-4
3.5.3	CDD/Repository	3-4
3.5.4	Cobol	3-4
3.5.5	DECforms	3-4
3.5.6	DECforms	3-4
3.5.7	ACMS Application	3-4
3.5.8	C	3-5
3.5.9	Cobol	3-5
3.5.10	TDMS Request Library	3-5
3.5.11	CDO	3-5
3.5.12	DECforms	3-5
3.5.13	TDMS Request Library	3-5
3.5.14	C	3-5
3.6	Command Procedures	3-5
3.6.1	SWDEV_CVT_RDB_TO.COM	3-5
3.7	Miscellaneous	3-6
3.7.1	REPLY/ENABLE	3-6
3.7.2	LSEDIT Buffer Initialization	3-6
3.7.3	LSEDIT Initialization Files	3-6

4 Known Limitations or Problems

4.1	Entry Points in MACRO-32	4-1
4.2	Entry Points in C	4-1
4.3	CVTMMS various	4-1

Index

Tables

2-1	Meta Classes added for V3.4	2-8
2-2	Meta Classes added for V3.3	2-8
2-3	Meta Classes added for V3.2	2-8
2-4	Meta Classes added for V3.1	2-9
2-5	Meta Classes added for V3.0	2-9
2-6	Meta Classes removed or renamed for V3.0	2-10
2-7	NET-APP-SUP License Usage	2-11
2-8	Old and New Generated SQL Script Names	2-15
2-9	Targets from File Name suffices	2-32

Send Us Your Comments

We welcome your comments on this manual or any SysWorks manual. If you have suggestions for improvements or find any errors, please indicate the chapter, section and page number (if available). Your input is valuable in improving future releases of our documentation.

You can send comments to us in the following ways:

- **Email**—<http://www.sysworks.com.au/contact.php>
- **Phone**—+61 (03) 9411 4411
- **FAX** —+61 (03) 9411 4499
- **Postal service**

SysWorks
Corpita Pty Ltd
15 Bedford Street
Collingwood VIC 3066
Australia

Preface

This manual provides release notes for SysWorks V3.5.

Intended Audience

This manual is intended for general circulation to all SysWorks installers and users.

Conventions

The following conventions are used in this document:

Conventions	Description
<code>Ctrl/X</code>	A sequence such as <code>Ctrl/X</code> indicates that you must hold down the key labeled <code>Ctrl</code> while you press another key or a pointing device button.
<code>[]</code>	In format descriptions, brackets indicate that whatever is enclosed is optional; you can select none, one or all of the choices. In system prompts indicates the default value which will be assumed if the Return key is pressed without first entering a value.
<code>{}</code>	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
<code> </code>	In format descriptions, vertical bars separate the options. If the options are enclosed in brackets (i.e. <code>[]</code>) you can select none, one or all of the choices. If the options are enclosed in braces (i.e. <code>{}</code>) you must choose one of the options listed.
<code>()</code>	In system prompts, parenthesis indicate the list of values one of which may be entered. The values are separated by a forward slash "/"
<code>...</code>	An elipsis indicates that a value within a range may be chosen or a syntax repeated. A range may be indicated by a pair of end values, or an end value and an end keyword. For example <code>Disk quota (0..unlimited)</code> indicates that the keyword <code>unlimited</code> may be used to represent the highest possible disk quota.
<i>italic text</i>	Italicized words and letters indicate that you should substitute a word or value of your choice.
UPPERCASE TEXT	Uppercase letters indicate the name of a command or routine.
<code>monospace text</code>	Normal monospace text indicates system prompts and output.
<code>bold monospace text</code>	Bold monospace text indicates user responses to system prompts.
<i><code>bold monospace italic text</code></i>	Bold monospace italic text indicates user responses to system prompts which need appropriate value substitution.

Conventions	Description
mouse	The term <i>mouse</i> is used to refer to any pointing device such as a mouse, a puck or a stylus.
MB1, MB2, MB3	MB1 indicates the left mouse button, MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (The buttons can be redefined by the user.)

Unless otherwise noted, all numeric values are represented in decimal notation.

Summary of Changes

1.1 Summary of Changes for V3.5

The following list provides a summary of changes made for V3.5:

- Section 2.1, Documentation Updates
- Section 2.2.12, OpenVMS Alpha 8.3

1.2 Summary of Changes for V3.4-1

1.3 Summary of Changes for V3.4

The following list provides a summary of changes made for V3.4:

- Section 2.2.20.1, V3.4

1.4 Summary of Changes for V3.3

The following list provides a summary of changes made for V3.3:

- Section 2.2.20.2, V3.3
- Section 2.4.16.8, CMS SHOW GENERATION - This command now supports the /ANCESTORS and /DESCENDANTS qualifiers. The /FORMAT qualifier now supports #T.
- Section 2.4.27.1, DEFINE - This command now supports the /[NO]QUOTE qualifier.
- Section 2.8.7, SWRK_EXTENDED_FILE_SEARCH - This routine has been enhanced to support the /SELECT qualifier.
- Section 2.8.9, SWRK_HANDLE_QUALIFIERS - This routine has been enhanced to allow a leading double ampersand to be interpreted as a single ampersand rather than as a symbol substitution operator.
- Section 3.1.1, Many Commands - Problems with /BEFORE and /SINCE on SysWorks™ commands have been fixed.
- Section 3.1.4.4, CMS INSERT GENERATION - Problem with not inserting generations has been fixed.
- Section 3.1.4.6, CMS RESERVE - Problem when reserving and variants has been fixed.
- Section 3.3.1, SWRK_HANDLE_QUALIFIERS - Problem with default dates has been fixed.

- Section 3.3.2, SWRK_MOVE_FILE - Problems with deleting have been fixed.

1.5 Summary of Changes for V3.2

The following list provides a summary of changes made for V3.2:

- Section 2.2.4, HTTP File Types
- Section 2.2.7, Message Help Support - Uses new file types, build rules and logical names.
- Section 2.2.8, Message Logging
- Section 2.2.15, Secure Task Actions
- Section 2.2.16, Security Model Changed
- Section 2.2.19, Synchronized Batch Jobs Status Message
- Section 2.2.20.3, V3.2
- Section 2.3.2, New Product Support - V3.2 supports Powerhouse from Cognos, Java, Purveyor from Process Software Corporation and TDMS Emulator from Praxa.
- Section 2.3.8, Java Support - Support for Java.
- Section 2.4.1, ACCEPT - Enhancements to /CONVERT and new /HELP qualifier.
- Section 2.4.2, BUILD - Enhancements to SETUP, SCAN and RULES phases and the /FROM[=location] qualifier supercedes /FROM_SOURCES.
- Section 2.4.3, BUILD and COMPILE - New /[NO]CROSS_REFERENCE qualifier.
- Section 2.4.16.2, ATTACH - Standard utility ATTACH command.
- Section 2.4.16.9, CONVERT/GENERATE - New /[NO]CHECK_SUBCONTEXT qualifier.
- Section 2.4.16.15, SHOW CONTEXT - New subcommand to show the current context.
- Section 2.4.23, SC - Now uses UTLTOOLS utility.
- Section 2.4.27.2, General - Previously internal utility is now documented.
- Section 2.4.29, WHO - A full show process utility.
- Section 2.7.11, appl_MSGHLP_LIB - A new application logical name.
- Section 2.8.1, SWRK_BINARY_SEARCH - A new routine to perform binary searches on tables.
- Section 2.8.3, SWRK_DISPLAY_HELP - A new routine to display OpenVMS help.
- Section 2.8.9, SWRK_HANDLE_QUALIFIERS - This routine has been enhanced to allow a leading ampersand to be interpreted as symbol substitution operator.
- Section 2.10.1, Improvements to All MMS Generators - Logical name translation attempted for all file based dependencies.
- Section 2.10.7, C - New logical name to control link time dependencies.

- Section 2.10.11, DECdocument - Improved handling of contents and indices.
- Section 3.1.2, ACCEPT - Problem with cursor position has been fixed.
- Section 3.1.4.7, CONVERT/GENERATE - More errors are now reported rather than causing image crash dumps.
- Section 3.5.2, CDD/Repository - Problems with CDD path detection has been fixed.
- Section 3.5.5, DECforms - Problems with copy statements has been fixed.

New and Modified Features

This chapter briefly describes the new and modified features in SysWorks.

2.1 Documentation Updates

V3.5 A new document SWRK035.RELEASE_NOTES may be found in SWRK_DOC_DIR: and in SYS\$HELP. These release notes now include a summary of changes for the current version in addition to the details of changes since V2.4-3.

Revised on-line help documents most SysWorks supported commands. Use HELP @SWDOCHLP for details.

SysWorks manuals are available using BNU and/or the Bookreader. If SysWorks Administrator is installed, a BNU/Bookreader replication utility is provided which scans BNU or Bookreader volumes and creates a master index in the SWADM_SFT_DIR: directory. This facility is used when SysWorks first mounts a BNU or Bookreader disk volume.

In light of this facility, there are now three ways to manage BNU/Bookreader access. These are:

- 1 Traditional method without using SysWorks documentation replicator. DECW\$BOOK and DECW\$BOOKSHELF logicals must be defined manually.
- 2 Manually use SysWorks documentation replicator and define DECW\$BOOK and DECW\$BOOKSHELF logicals to refer to the replication results.
- 3 Install SysWorks at the turnkey level, in which case the replicator is automatically run during installation and the DECW\$BOOK and DECW\$BOOKSHELF logicals are defined during installation and at system boot time.

2.1.1 Traditional Method

In the traditional method, DECW\$BOOK and DECW\$BOOKSHELF logical names should be defined in the system startup procedures using commands such as:

```

$ DEFINE/SYSTEM DECW$BOOK -
  DISK$VAXDOCSEP961: [DECW$BOOK] , -
  DISK$VAXDOCSEP962: [DECW$BOOK] , -
  DISK$VAXDOCSEP963: [DECW$BOOK] , -
  DISK$VAXDOCSEP963: [DECW$BOOK] , -
  SWRK_DOC_DIR: , -
  SWEXA_DOC_DIR: , -
  SWPUB_DOC_DIR: , -
  SYS$SYSROOT: [DECW$BOOK]
$ DEFINE/SYSTEM DECW$BOOKSHELF -
  DISK$VAXDOCSEP961: [DECW$BOOK] LIBRARY.DECW$BOOKSHELF; , -
  SWRK_DOC_DIR: LIBRARY.DECW$BOOKSHELF; , -
  SYS$SYSROOT: [DECW$BOOK] LIBRARY.DECW$BOOKSHELF;

```

In the above example, the logical names DISK\$VAXDOCSEP96*n* are the standard default logical names defined by a MOUNT command.

Note also that the trailing semi-colons on the DECW\$BOOKSHELF equivalences are required to stop the bookreader opening multiple copies of the same library. Typically these commands may be placed in the SWRK_LCL_DIR:site_PRE_STARTUP.COM command procedure.

2.1.2 Replicator Method

If SysWorks documentation replicator is used manually (either by using SWADM_SFT_DIR:SWADM_IMPORT_DISK_TAPE_INFO.COM or the DIRTOOLS DOCUMENTATION/REPLICATE command), the DECW\$BOOK and DECW\$BOOKSHELF logical names should be defined in the system startup procedures using commands such as:

```

$ DEFINE/SYSTEM DECW$BOOK -
  SWRK_DOC_DIR: , -
  SWEXA_DOC_DIR: , -
  SWPUB_DOC_DIR: , -
  SYS$SYSROOT: [DECW$BOOK]
$ DEFINE/SYSTEM DECW$BOOKSHELF -
  SWADM_DAT_DIR: LIBRARY.DECW$BOOKSHELF; , -
  SWRK_DOC_DIR: LIBRARY.DECW$BOOKSHELF; , -
  SYS$SYSROOT: [DECW$BOOK] LIBRARY.DECW$BOOKSHELF;

```

Note that using SysWorks at the turnkey level should generate the same results as the above commands.

2.2 Concepts

This section briefly describes new and modified concepts provided in SysWorks.

2.2.1 Alternative Installation Directory

V2.5-1 If SysWorks is installed at the public level, the installation procedure will ask for an installation device. By default, SYS\$SYSDEVICE: is assumed which results in SysWorks being placed in the standard system tree. If another disk is specified, the appropriate directory structures are placed in the root of the indicated disk.

2.2.2 Application Environment Scope

V2.4-3 SysWorks now supports a private versus common scope for compilations and builds. When an application environment scope is common, all developers share the standard directories such as work, library and software. When the environment is scope private, each developers actions take place in their work and runtime directories. This is achieved by making the logical names associated with the standard directories into search lists, with the developers work or runtime directory as the first equivalence, and the standard shared directory as the second.

2.2.3 Error Handling

V3.1 SysWorks has significantly improved internal error handling. All crashes now generate a log file in addition to any output to the normal SYS\$ERROR and SYS\$OUTPUT files. The log file name is based on the image in which the error occurred. For example, if a DEVTOOLS command failed, a log file of the name SWDEV_DEVTOOLS.LOG would be created. If a problem occurred within LSEDIT, an LSEDIT.LOG file would be created.

The log file output now includes a call frame dump for Alpha in addition to VAX. Earlier versions only did this on VAX. Note that this call frame dump is generated even if the image is linked /NODEBUG/NOTRACEBACK.

Applications can use these facilities within SysWorks by calling SWRK_ESTABLISH to establish a universal exception handler. This requires SysWorks to be present on production systems. The SysWorks Base product includes these facilities.

2.2.4 HTTP File Types

V3.2 SysWorks now supports various HTTP file types. Although these were supported using the _CPY and/or _SRC file type suffixes, in practice use of these file types makes it difficult to develop HTTP pages. The file types currently supported include:

.GIF
.HTIMAGE
.HTM
.HTML
.HTMLS
.HTMLX
.IMAGEMAP

SysWorks supports using these file types for both source and target files.

2.2.5 Installation Level

V3.0 SysWorks now supports three installation levels. These are:

- Public
- System
- Turnkey

SysWorks no longer supports the private installation level introduced in V2.4-3. See the SysWorks Installation Guide for more details.

2.2.6 Library Module Deletion

- V3.0** SysWorks now deletes modules in object, shareable image and text libraries in addition to files as a consequence of appropriate /TIDY qualifiers. See notes on the BUILD, DEVTOOLS CMS FETCH, DEVTOOLS CMS RESERVE and DEVTOOLS DELETE commands for details.

2.2.7 Message Help Support

- V3.2** SysWorks now supports HELP/MESSAGE sources and data files. The file types used are .HLPMSG and .HLPMSG\$DATA respectively. Because the .MSGHLP\$DATA file type is not an OpenVMS library, and hence MMS cannot directly support it, an intermediate tag file of file type .TAG_MSGHLP is used to indicate to MMS that the .MSGHLP source has been inserted into the .MSGHLP\$DATA file.

An application logical name of the form *appl_MSGHLP_LIB* must be used to identify the application message help data file. The SysWorks convention for the specification of this file (i.e. the equivalence of the logical name) is:

```
appl_DOC_DIR: applMSGHLP.MSGHLP$DATA.
```

The SysWorks message help data file is now incorporated into the system message help search list MSGHLP\$LIBRARY . If this logical is not defined, SysWorks defines it as a search list of SYS\$HELP:MSGHLP\$LIBRARY.MSGHLP\$DATA and SWRK_MSGHLP_LIB. If it is already defined and SWRK_MSGHLP_LIB is not yet included, SysWorks adds SWRK_MSGHLP_LIB as the last item in the search list.

2.2.8 Message Logging

- V3.2** SysWorks now uses a utility to display messages from DCL command procedures rather than simply writing text to SYS\$OUTPUT. This utility uses the SysWorks message logging routines which in turn use the conventional OpenVMS system service SYS\$PUTMSG to access the SysWorks message shareable image in order to retrieve and display the text of the message.

This results in a much tighter integration of DCL command procedures with executable images and produces more consistent error messages.

2.2.9 Multi-architecture Support

- V3.1** SysWorks now supports a /COMBINED qualifier which allows a single BUILD task to generate software for both architectures. It also enhances various DEVTOOLS commands which deletes files and library modules by searching the directories associated with both architectures for items to delete. See notes on the BUILD, DEVTOOLS CMS FETCH, DEVTOOLS CMS RESERVE and DEVTOOLS DELETE commands for details.

- V3.0** SysWorks now supports multi architecture development for OpenVMS on VAX and Alpha. In order to support this, various directories now have architecture dependent suffices. The following table indicates the old directories and the corresponding new directories:

Old	New
LIB	LIB, LIB-ALPHA, LIB-VAX

Old	New
SFT	SFT-ALPHA, SFT-VAX
TST	TST-ALPHA, TST-VAX

The application logical names have equivalences which use the architecture specific directories. Note that the LIB directory is retained for architecture independent intermediate files such as CDD/Repository tag files. A new directory code of AIL is used for this architecture independent directory. Thus FIN_LIB_DIR might have an equivalence of DISK_DEV8:[FIN.LIB-ALPHA] whereas FIN_AIL_DIR might have an equivalence of DISK_DEV7:[FIN.LIB].

Another feature of multi-architecture support is architecture specific sources. These have a suffix of the form -ALPHA or -VAX in the file name. In order to comply with the MMS implicit rules convention, a copy of such sources are made into the library directory before any compilation or other build action takes place. For example the source FIN_SUB_PROG_A.COB causes an MMS commands such as the following to be generated:

```
FIN_LIB_DIR:FIN_SUB_PROG_A.OBJ depends_on -
  FIN_WRK_DIR:FIN_SUB_PROG_A.COB
```

If the source was architecture specific (eg. was named FIN_SUB_PROG_A-ALPHA.COB) the following script would be generated:

```
FIN_LIB_DIR:FIN_SUB_PROG_A.COB depends_on -
  FIN_WRK_DIR:FIN_SUB_PROG_A-ALPHA.COB
  $(copy) $(mms$source) $(mms$target)
FIN_LIB_DIR:FIN_SUB_PROG_A.OBJ depends_on -
  FIN_LIB_DIR:FIN_SUB_PROG_A.COB
```

In the case indicated above, the internal name (eg. for Cobol the program-id) should remain without the architecture suffix.

2.2.10 Multi-variant Support

V3.0 SysWorks now supports multi variant environments for development and development testing. The CONTEXT command has been extended to incorporate version parameters on the APPLICATION and ENVIRONMENT sub-commands and a new VARIANT sub-command. A new convenience command VARIANT has been added.

The variant directory is placed between the application directory and the 'second level' directory. For example DISK_DEV:[FIN.WRK] would be the most stable and default development area, whereas DISK_MNT:[FIN.VAR-B.WORK] would be for developing new code referred to as variant B.

The CMS class has the version code appended with an underscore to the primary class name. For example FIN_MNT would be the CMS class for the latest and default version, whereas FIN_MNT_V021 would be for support of the older V2.1 version.

2.2.11 Multi-version Support

V2.5-1 SysWorks now supports multi version environments for maintenance and maintenance testing. The CONTEXT command has been extended to incorporate version parameters on the APPLICATION and ENVIRONMENT sub-commands and a new VERSION sub-command. A new convenience command VERSION has been added.

At this time the format of version numbers is Vn.m. Note that some massaging of the version number takes place when converting it to a version code, for example V2.1 becomes V021.

The version directory is placed between the application directory and the 'second level' directory. For example DISK_MNT:[FIN.WORK] would be the latest and default maintenance version, whereas DISK_MNT:[FIN.V021.WORK] would be for support of the older V2.1 version.

The CMS class has the version code appended with an underscore to the primary class name. For example FIN_MNT would be the CMS class for the latest and default version, whereas FIN_MNT_V021 would be for support of the older V2.1 version.

2.2.12 OpenVMS Alpha 8.3

V3.5 SysWorks now supports OpenVMS Alpha Version 8.3 as well as OpenVMS VAX Version 7.3.

2.2.13 OpenVMS Alpha

V3.0 SysWorks now supports OpenVMS alpha as well as OpenVMS VAX. As part of this support, multi-architecture application development is also supported. Also, some MMS script generators create slightly different syntax as required by the architecture differences.

2.2.14 Oracle Rdb Version

V3.1 This release of SysWorks includes support for Oracle Rdb V6.1 in addition to DECrdB V6.0A, V5.0A and V4.2.

V2.6 This release of SysWorks includes support for multi-variant DEC Rdb installations. Previous versions of SysWorks changed some behaviour based upon which version of DEC rdb was installed. These changes will now take place based on which version of DEC Rdb has been selected with the SYS\$LIBRARY:DECRDB_SETVER or SYS\$LIBRARY:RDBVMS_SETVER command procedure.

2.2.15 Secure Task Actions

V3.2 The SysWorks secure tasks now take an argument indicating whether files are to be secured using the default action (i.e. SET SECURITY/DEFAULT) or an explicit action (i.e. SET SECURITY/ACL=(...)/OWNER=.../PROTECTION=(...)).

The advantage of the default secure action is that it is faster and does not change the modification dates of the secured files. This preservation of modification dates stops an incremental build following a secure task turning into an effective full build.

The advantage of the explicit secure action is that it is more thorough. It was also the action used in earlier versions of SysWorks. Note that an incremental build following a secure task using the explicit action effectively becomes a full build since all sources appear to be new.

By default the default action is used.

2.2.16 Security Model Changed

V2.5 SysWorks no longer includes the `OPTIONS=PROTECTED` clause in ACLs. This change is made to allow the use of the `/DEFAULT` qualifier (and hence the `DEFAULT` action as indicated above). The `SET SECURITY/DEFAULT` command only works for ACEs which do not have the `PROTECTED` option set.

V2.5 When SysWorks is installed at or below the public installation level, a series of symbols is used to indicate identifiers to be used to provide security for objects which need ACLs.

When SysWorks is installed at or above the system level, these symbols are not necessary, since SysWorks registers and uses identifiers to achieve this end.

In previous versions of SysWorks the value of the `ACE_MGR` symbol represented the sole identifier which was thus used.

This symbol has been superceded by the `ID_MGR` symbol. Although `ACE_MGR` may continue to be used, it may not be supported in a future version of SysWorks. Two other symbols, `ID_USR` and `ID_REP` have been added to support a three layer ACL. The symbols are used to grant the following access rights:

Symbol	Rights to associated identifier
<code>ID_MGR</code>	<code>READ+WRITE+EXECUTE+DELETE+CONTROL</code>
<code>ID_USR</code>	<code>READ+WRITE+EXECUTE+DELETE</code>
<code>ID_REP</code>	<code>READ+EXECUTE</code>

These symbols would normally be defined in an application environment's `ENTER.COM` and deleted in its `EXIT.COM`.

2.2.17 SPECIFIC Scope

V2.6 This release of SysWorks includes full support for the `SPECIFIC` scope for application development and maintenance. When using the `SPECIFIC` scope, developers and maintainers compile and build within their own sub-directories, while still sharing the application `COMMON` area for sources, objects and images etc. which they are not modifying themselves.

2.2.18 Subprocess Editor

V2.4-3 The default `EVE` or `LSEEDIT` editor is now kept as a subprocess by default. To disable this feature, remove the `ED*IT`, `EVE` and/or `LSE*DIT` components from the `SWRK_SPAWN_COMMANDS` logical name. Note that if a logical name needs to be empty, it has to consist of a space.

2.2.19 Synchronized Batch Jobs Status Message

V3.2 Batch jobs which can be synchronized with another batch job (i.e. those submitted by commands which support the /SYNCHRONIZE qualifier - this includes the BUILD, COMPILE and DO commands) now set a restart message indicating their synchronizing state. This is similar to the executing phase messages which BUILD jobs provide. This feature allows a user or operator to easily distinguish batch jobs which are waiting on other batch jobs to complete.

2.2.20 Meta object Model

2.2.20.1 V3.4

V3.4 SysWorks V3.4 adds support for the new meta classes listed in Table 2-1.

Table 2-1 Meta Classes added for V3.4

Name	Code	Description
Computer_Node	CPND	Computer_Node
Computer_Type	CPTY	Computer_Type
Node	NDND	Node

2.2.20.2 V3.3

V3.3 SysWorks V3.3 adds support for the new meta classes listed in Table 2-2.

Table 2-2 Meta Classes added for V3.3

Name	Code	Description
Node_Class	NDCL	Node Class
Node_Type	NDTY	Node Type

2.2.20.3 V3.2

V3.2 SysWorks V3.2 adds support for the new meta classes listed in Table 2-3.

Table 2-3 Meta Classes added for V3.2

Name	Code	Description
Architecture	ARCH	Architecture
File_Type	FLTY	File Type
Group_User	GRUS	Group User
IP4_Address	INA4	IP4 Address
Internet_Domain	INDM	Internet Domain
IP4_Subnet	INSN	IP4 Network/Sub-net
Name_Model	NMMD	Name Model
NODE_class	NDCL	NODE class
Oper_Sys	OPSY	Operating System
Playform	OSPL	Platform (Operating System Architecture)
Property	PROP	Property

Table 2–3 (Cont.) Meta Classes added for V3.2

Name	Code	Description
Property_Type	PPTY	Property Type
UNIX_Username	UNXU	UNIX Username
UNIX_Username_ Node	UUND	UNIX Username Node
Variable_Type	VBTY	Variable Type

2.2.20.4 V3.1

V3.1 SysWorks V3.1 adds support for the new meta classes listed in Table 2–4.

Table 2–4 Meta Classes added for V3.1

Name	Code	Description
Directory_Profile	PFDR	Directory Profile
Profile_Item	PFIT	Profile Item

2.2.20.5 V3.0

V3.0 SysWorks V3.0 adds support for the new meta classes listed in Table 2–5.

Table 2–5 Meta Classes added for V3.0

Name	Code	Description
Appl_envr_User	AEUS	Application Environment User
Base	BASE	Base Object
DECnet_Address_Number	DNNN	DECnet address Number
Device_Type	DVTY	OpenVMS Device Type
Disk_Device_Type	DKDT	Disk Device Type
Disk_Media	DKMD	Disk Media
Disk_Media_Format	DKMF	Disk Media Format
Disk_Media_Type	DKMT	Disk Media Type
Location_Type	LCTY	Location Type
Member	MBER	Member
Member_Category	MBCT	Member Category
Message	MSAG	Message
Method	MTHD	Method
Method_Package	MTPK	Method Package
NODE_type	NDTY	NODE type
Owner	OWNR	Owner
PAK	LPAK	Product Authorisation Key
Product	LYPR	Layered Product
Relationship	RELN	Meta object Relationship
Security_Model	ACLX	Security Model

Table 2–5 (Cont.) Meta Classes added for V3.0

Name	Code	Description
System_Job	SYJB	System Job
Tape_Device	MTDV	Tape Device
Tape_Device_Type	MTDT	Tape Device Type
Tape_Media	MTMD	Tape Media
Tape_Media_Type	MTMT	Tape Media Type
Tape_Pool	MTPL	Tape Pool
Tape_Volume	MTVL	Tape Volume
VMS_Username_Node	VUND	OpenVMS Username Node

SysWorks V3.0 removes support for or renames the meta classes listed in Table 2–6.

Table 2–6 Meta Classes removed or renamed for V3.0

Name	Code	Reason
Appl_envr_Node	AEND	Converted to VMS_Username_Node
Group_Node	GRND	Converted to VMS_Username_Node
Product	LYPD	Renamed to LYPR

2.2.21 Test Directory

V2.6 This release of SysWorks supports an application environment test directory. Like other application directories, this directory can be adjusted using the SDC_TST and DIR_TST symbols. The default value of the SDC_TST symbol is TST.

The feature of the test directory is that all targets beginning with *appl_TST* (or *appl_sdc-tst* if a site specific SDC_TST symbol is used) will be placed in the test directory rather than the standard software directory.

To disable this feature, set the SDC_TST symbol to a space.

As part of this new feature, SysWorks now has a test directory SWRK_TST_DIR which contains a number of tests of SysWorks features. These tests consist of images and DCL command procedures which may be executed at any time.

2.2.22 Utility Logicals Scope

V2.5-1 If SysWorks is installed at the public level, the various utility logicals supported in SysWorks such as EDTINI may be defined at the system or user level. The installation procedure ask a question regarding this. The response must be either SYSTEM or USER. Note that with a USER response, initial user logins after a system boot will be slower.

2.2.23 VARIANT Context

V3.0 This release of SysWorks includes support for variant (i.e. concurrent or parallel) development of applications.

To allow specific environments to use this context in a public installation, commands similar to the following should be used to define the appropriate control logical names:

```
$ DEFDAT :=> DEFINE/NOLOG/TABLE=LNm_swrk_DATABASE
$ DEFDAT SWRK_ENVR_APPL "SCOP=COMMON\VACL=0\VRNT=0\VRSN=0\WORK=0"
$ DEFDAT SWRK_ENVR_DEV "SCOP=SPECIFIC\VACL=0\VRNT=1\VRSN=0\WORK=1"
$ DEFDAT SWRK_ENVR_DTST "SCOP=COMMON\VACL=0\VRNT=1\VRSN=0\WORK=1"
$ DEFDAT SWRK_ENVR_FDEV "SCOP=SPECIFIC\VACL=0\VRNT=1\VRSN=0\WORK=1"
$ DEFDAT SWRK_ENVR_MNT "SCOP=SPECIFIC\VACL=1\VRNT=0\VRSN=1\WORK=1"
$ DEFDAT SWRK_ENVR_MTST "SCOP=COMMON\VACL=1\VRNT=0\VRSN=1\WORK=1"
$ DEFDAT SWRK_ENVR_PROD "SCOP=COMMON\VACL=0\VRNT=0\VRSN=0\WORK=0"
$ DEFDAT SWRK_ENVR_TRNG "SCOP=COMMON\VACL=0\VRNT=0\VRSN=0\WORK=0"
```

These commands are typically defined in `SWRK_LCL_DIR:site_PRE_STARTUP.COM`. The new `SCOP` attribute indicates whether only the `COMMON` scope is permitted or the `SPECIFIC` scope is allowed as well. The `VACL`, `VRSN` and `WORK` components were described in previous release notes.

2.3 Layered Products

This section briefly describes new and modified layered product support in SysWorks.

2.3.1 New License Recognition

V2.6 The `NET-APP-SUP` license are now recognised as valid licenses for various products. See Table 2–7 for details.

Table 2–7 NET-APP-SUP License Usage

Product	Minimum NET-APP-SUP License
ACA Services	250
ACMS	400
DCPS	250
DEC TCP/IP Services	200
DECforms	250
DECtrace	400
DQS	250

2.3.2 New Product Support

V3.2 Support has been added or enhanced for the following layered products and versions:

- BNU Bookshelf Navigation Utility. See also Section 2.1.
- HyperHelp (from Bristol Technology Inc).
- Java.
- Powerhouse (from Cognos).
- Purveyor (from Process Software Corporation).

- TDMS Emulator, TDMS Emulator Developer and TDMS to DECforms Converter (from Praxa Limited).

V3.0 Support has been added or enhanced for the following layered products and versions:

- All-In-1 V3.1
- DDS
- DEC Mailworks V1.2, Mailworks V1.3A
- Microsoft Windows V3.1, V3.11
- Microsoft Windows for Workgroups V3.11
- Microsoft Word for Windows V6.0
- MS-DOS V6.21
- ObjectBroker V2.5
- Oracle CDD/Repository V6.1
- Oracle Rdb for OpenVMS V6.1
- Pathworks V5.0, V5.1
- TeamLinks V2.1
- TeamRoute V1.0

V2.6 Support has been added for the following layered products:

- ADA
- Basic
- Bookreader
- C++
- DEC File Optimizer (DFG)
- DEC Rdb for OpenVMS VAX V4.2, V5.1 and V6.0
- DEC Mailworks
- DECdocument
- DECprint Supervisor for OpenVMS (DCPS)
- Pascal
- POSIX
- VUIT

2.3.3 ACA Services Support

V2.5 SysWorks now supports ACA Services. It provides MMS generators for .COL and .CRL files, an ACAS convenience command and a DECwindows/Motif Application pulldown menu item.

2.3.4 DECdocument

- V2.6-1** All SysWorks MMS generators for DECdocument which previously used manual designs now use software designs. SysWorks now provides an enhanced software design file, and no longer provides any manual design files.

2.3.5 DECset V12.1

- V3.1** Specific SysWorks versions are linked against specific DECset versions. The following table lists these combinations:

SysWorks Version	DECset Version
V3.1	V12.1
V3.0	V12.0
V2.5 - V2.6-1	V11.0

If SysWorks is configured to use DECset, they must be of the appropriate version. In a future version of SysWorks, the installation procedure will link against the version of DECset which is present on the installation computer. Special arrangements can be made for customers who need features of a newer version of SysWorks with an older version of DECset i.e. where the DECset version cannot be updated.

2.3.6 DECset V12

- V3.0** SysWorks now provides enhanced Session Manager and FileView support using the new DECset pulldown menu.

The SysWorks™ CONTEXT command and the SysWorks Administrator Application Management tasks now support DECset environment database and contexts.

2.3.7 DECset V11

- V2.5** SysWorks now provides enhanced LSEDIT support including the LSEDIT Another File item in the Utilities pulldown menu. The reuseable DECset Debugger is also provided in the Utilities pulldown menu.

2.3.8 Java Support

- V3.2** SysWorks now supports Java. It provides MMS generators for .CLASS and .JAVA files.

2.3.9 ObjectBroker Support

- V3.0** SysWorks now supports ObjectBroker. It provides MMS generators for .COL, .IDL, .IML and .MML files, an command and a DECwindows/Motif Application pulldown menu item. Note that SysWorks cannot support ObjectBroker and ACA Services on the same computer node - if both are present it supports ObjectBroker rather than ACA Services.

2.4 Commands

This section briefly describes new and modified commands provided in SysWorks.

2.4.1 ACCEPT

- V3.2 The ACCEPT command has enhancements to the /CONVERT qualifier and also has a new /HELP qualifier. See HELP ACCEPT for more details.

2.4.2 BUILD

- V3.2 The SETUP phase now has default actions which are executed if no application specific SETUP procedure is present. These default rules ensure that the standard ENTER.COM, EXIT.COM and LOGICALS.COM procedures exist, are up to date and have been executed.

- V3.2 The SCAN and RULES phases default actions have been enhanced to include actions for the generated source directory if present.

In previous versions of SysWorks Developer, applications that generated sources needed to provide application specific SCAN and RULES procedure which executed SWDEV_SFT_DIR:SWDEV_BUILD_phase without a parameter for the normal source directory and then again with an appropriate parameter for the generated source directory.

Now, the default action (i.e. when there is no application specific procedure) works as though these two calls were made. Thus, many applications may now remove their application specific procedures for these phases.

- V3.2 The /FROM_SOURCES qualifier has been replaced by the /FROM[=location] qualifier. The location value can be LIBRARY, SOURCE or WORK. See HELP BUILD /FROM for more details. The default location of SOURCE yields the same effect as the old /FROM_SOURCES qualifier.

- V3.2 The /RELATED qualifier is now implemented. This allows related applications to be built in a single job.

- V3.1 The /COMBINED qualifier is now implemented. This allows the Alpha and VAX versions of an application to be built in a single task when the BUILD command is executed on an Alpha.

A consequence of this is that the default names for log files generated during non combined builds have an architecture suffix. For example BUILD/NOCOMBINED will by default generate a BUILD-VAX.LOG log file on a VAX.

- V3.1 A new GENERATED default phase has been added. This phase is executed between the SETUP and SCAN phases. The default values when /PHASES=ALL is specified (or defaulted to when /FETCH is specified) is now SETUP, GENERATE, SCAN, RULES, DESCRIP.

As a consequence of the introduction of this new phase, the SCAN and RULES phases will now automatically generate a second set of MMS and/or SQL scripts based on the sources in the generated source directory (directory code GEN). For example, in the FIN application, FINMMS.TLB would be used to store the source level MMS scripts and FIN_DEPENDENCIES.MMS would be generated as the combined script. If there were sources in the FIN_GEN_DIR: directory, FINMSGEN.TLB would be used to hold the individual MMS scripts and FIN_DEPENDENCIES_GEN.MMS would be generated as an additional large script.

- V3.0 The RULES and DESCRIP phases now jointly generate the RDO and/or SQL scripts for creating a DEC Rdb database which used to be generated during the SCAN phase. The file names of these generated files have now changed. See Table 2-8 for a comparison of the old and new names.

Table 2–8 Old and New Generated SQL Script Names

Old Script Name	New Script Name
<i>appl_CONRAINTS</i> .SQL	No longer generated
<i>appl_DOMAINS</i> .SQL	<i>appl_SCHEMA_DOM</i> .SQL
<i>appl_INDICES</i> .SQL	<i>appl_SCHEMA_IDX</i> .SQL
<i>appl_TABLES</i> .SQL	<i>appl_SCHEMA_TBL</i> .SQL
<i>appl_TRIGGERS</i> .SQL	<i>appl_SCHEMA_TRG</i> .SQL

V2.6 The /KIT qualifier has been changed to /KITBLD and the KIT phase has been changed to KITBLD. This is to increase consistency with the VMSOIINSTAL conventions.

The BUILD command SCAN phase now generates an *appl_DOCUMENTATION*.MMS script in the library directory. This script contains a target called DOCUMENTATION which depends on all the documentation targets.

Application DESCRIP.MMS files should be changed to reflect this by adding the DOCUMENTATION source to the ALL target and including the library directory *appl_DOCUMENTATION*.MMS script.

See SWRK_SFT_DIR:DESCRIP.TEMPLATE for an example.

V2.5 The BUILD/FETCH qualifier now uses the DEVTOOLS CMS FETCH command with an empty remark so that CMS history is not full or BUILD comments. If the BUILD is being performed in the COMMON scope, BUILD/FETCH uses the DEVTOOLS CMS FETCH/TIDY qualifier which deletes intermediate and target files associated with the element being fetched in a similar fashion to the BUILD/CMS qualifier.

The BUILD/PHASE=ALL no longer includes the KIT phase. The BUILD/KIT qualifier now provides this functionality.

The new BUILD/TIDY qualifier cleans up a developers specific areas by taking the following steps:

- For each file in the common work directory which the developer does not have reserved any corresponding files in the developers specific work, library or software directories are deleted.
- All .TAG_CDD files are deleted from the developers library directory.
- All objects in the developers specific area in the application environment CDD/Plus dictionary or CDD/Repository repository are deleted.
- The existing developers specific object library is deleted and an empty one created.

This qualifier should be used after a BUILD/COMMON is performed to remove unnecessary duplicate specific objects and targets.

2.4.3 BUILD and COMPILE

V3.2 The BUILD and CHANGE commands now support the /[NO]CROSS_REFERENCE qualifier for appropriate compilers and the Linker. Refer to the compiler and Linker documentation for details.

V3.0 The BUILD and CHANGE commands now support the /[NO]MACHINE, /[NO]MEMBER_ALIGNMENT and /[NO]OPTIMIZE qualifiers for compilers which support them.

The /DEBUG qualifier now defined a C macro by applying a /DEFINE=DEBUG=1 qualifier to C and C++ commands.

2.4.4 BUILD, CONTEXT APPLICATION and CONTEXT ENVIRONMENT

V2.4-3 These commands now support /COMMON and /SPECIFIC qualifiers to override the default scope when entering an application environment.

2.4.5 CHANGE

V2.5 The CHANGE command is now a convenience command for DEVTOOLS CHANGE rather than being a standalone utility.

2.4.6 CHGCTL

V2.6 There is a totally new change control system. The existing change control system is now known as the version control system (see VSNCTL below). This menu includes options to goto the source, testing and version control menus. This menu is available as **Manage** ⇒ **Change** from the session manager menu bar.

2.4.7 COMMON

V2.4-3 Sets the application environment to the common scope which will cause compilations and builds to occur in the shared directories.

2.4.8 COMPARE

V2.5 The COMPARE command is now a convenience command for DEVTOOLS DIFFERENCES/DATE.

2.4.9 CONTEXT

V3.2 The CONTEXT command now supports a RESET sub-command which resets the current context. This is useful to ensure that standard procedures such as EXIT.COM, LOGICALS.COM and ENTER.COM have been executed. It is used by the default BUILD SETUP phase if any of these procedures have been updated.

V2.6-1 When an application environment or group logical name table does not exist, the CONTEXT command no longer uses process privileges to create a system wide logical name table for an application environment or group. In a future release of SysWorks the CONTEXT command will always create system wide logical name tables for applications environments and groups. Until this time, process logical name tables will always be created.

V2.5-1 The new CONTEXT command supercedes the existing APPLICATION, ENVIRONMENT, GROUP, HOME and USER commands. These commands are now convenience commands based on the CONTEXT command. The summary of convenience commands based around the CONTEXT command includes:

Command	Equivalence
APPLICATION	CONTEXT APPLICATION app-cod env-cod [vsn-cod]
ENVIRONMENT	CONTEXT ENVIRONMENT env-cod [vsn-cod]
GROUP	CONTEXT GROUP grp-cod
HOME	CONTEXT HOME
USER	CONTEXT USER [usr-cod]
VERSION	CONTEXT VERSION vsn-cod

2.4.10 CVTMMS

V3.0 The CVTMMS convenience command is now implemented using the following command:

```
DEVTOOLS CONVERT/GENERATE/DEFAULT=TARGET=.MMS_INC.
```

2.4.11 CVTMMS

V2.5 The CVTMMS convenience command converts a source into an MMS script.

2.4.12 CVTRDB

V2.5 The CVTRDB convenience command converts an Rdb/OpenVMS database into a set of sources.

2.4.13 DEC Rdb Precompiler Commands

V2.4-3 The following is now the list of Rdb/VMS precompiler commands:

Command	Equivalence
RBAS	\$RDBPRE/BASIC
RCOB	\$RDBPRE/COBOL
RDMLC	\$RDML/CC
RDMLPAS	\$RDML/PASCAL
RFOR	\$RDBPRE/FORTRAN
SADA	\$SQL\$PRE/ADA
SBAS	\$SQL\$PRE/BASIC
SCC	\$SQL\$PRE/CC
SCOB	\$SQL\$PRE/COBOL
SFOR	\$SQL\$PRE/FORTRAN
SPAS	\$SQL\$PRE/PASCAL
SPLI	\$SQL\$PRE/PLI

The RDBCBOBOL command has been removed.

2.4.14 DIRTOOLS

V3.2 The DIRTOOLS utility is provided with SysWorks Administrator. It supports the following specific commands:

```
DOCUMENTATION/REPLICATE
```

2.4.15 DO

V2.5-1 The DO command has the following new qualifiers:

- /INTERACTIVE
- /PRODUCT=prd-cod
- /REMOTE=(computer-node-list[,...])
- /SYMBOLS
- /TERMINAL
- /WINDOW

It no longer supports the following qualifiers:

- /CLUSTER
- /[NO]DEFAULTS
- /[NO]MEMORY
- /COMPUTER_NODE
- /SAVE
- /UNSAVE

2.4.16 DEVTOOLS

This section briefly describes new and modified sub-commands of the DEVTOOLS command.

V2.6 A series of new commands has been added to the DEVTOOLS utility. These commands are used for invoking and controlling the invoking of standard layered product utilities. At this time, ACMS (ADU), CDD/Repository (CDO) and TDMS (RDU) are supported. Specifically they are designed to allow a subprocess running the layered product utility to be kept. This has the advantage of eliminating the access or bind time normally associated with running these utilities. This functionality is now used by the BUILD command, and greatly increases the speed of building when many of the actions involve one of the layered products.

The CMS commands which supported the /CLASS qualifier no longer do so. This functionality is achieved using the /GENERATION qualifier. Furthermore /GENERATION qualifier now supports a search list of generations where appropriate.

V2.5 SysWorks V2.5 introduces the DEVTOOLS utility. This utility provides enhanced functionality for DCL commands, CMS and some command of its own.

A typical enhanced DCL command is DEVTOOLS COPY/NODUPLICATES which appends a set of files into one output file, but does not duplicate any of the files copied. This is useful if a search list is used so that files from later areas are not copied if a file from an earlier area has been copied. This is used in the creation of the *appl_APPENDS.MMS* and assists when this is being built in a developer specific area.

The CMS front end provides SRCCTL like functionality in an image rather than a command procedure. Note that not all of the functionality in SRCCTL or CMS itself is yet provided. In a future release of SysWorks the DEVTOOLS utility

will provide a complete set of CMS commands including those which it does not enhance.

Specific DEVTOOLS commands include DEVTOOLS DIFFERENCES/DATES which is used by the BUILD/FETCH command to fetch generations in the CMS library which are newer than those in an application's work directory.

The DEVTOOLS utility also provides a command mode with command recall etc.

2.4.16.1 ADU

V2.6 The DEVTOOLS ADU[/KEEP] command invokes the ACMS ADU utility.

2.4.16.2 ATTACH

V3.2 The DEVTOOLS ATTACH command switches the users terminal to another process in a similar fashion to the ATTACH command in most Digital utilities.

2.4.16.3 CDO

V2.6 The DEVTOOLS CDO[/KEEP] command invokes the CDD/Repository CDO utility.

2.4.16.4 CMS FETCH

V2.6-1 The DEVTOOLS CMS FETCH command now supports the /MERGE qualifier. See HELP DEVTOOLS CMS FETCH /MERGE for details.

2.4.16.5 CMS PROMOTE

V2.6 There is a new DEVTOOLS CMS PROMOTE command for promoting a generation between a developers private class and an environments command class. See HELP DEVTOOLS CMS for full details on the supported sub-commands and qualifiers.

2.4.16.6 CMS REPORT

V2.6-1 There is a new DEVTOOLS CMS REPORT command for generating a report about a CMS library. See HELP DEVTOOLS CMS REPORT for full details on the supported qualifiers.

2.4.16.7 CMS RESERVE

V2.6-1 The DEVTOOLS CMS RESERVE command now supports the /PROMOTION_CHECK qualifier. See HELP DEVTOOLS CMS RESERVE /PROMOTION_CHECK for details.

2.4.16.8 CMS SHOW GENERATION

V3.3 The DEVTOOLS CMS SHOW GENERATION command now supports the /ANCESTORS and /DESCENDANTS qualifiers. The /FORMAT qualifier now supports #T to indicate substitution of the generation creation date and time. See HELP DEVTOOLS CMS SHOW GENERATION for details.

2.4.16.9 CONVERT/GENERATE

- V3.2 The DEVTOOLS CONVERT/GENERATE command now supports a /CHECK_SUBCONTEXT qualifier (which is assumed by default). This qualifier indicates whether to check that the context directory subcontext is valid (eg. the library directory exists). The REDUCE convenience command now includes /NOCHECK_SUBCONTEXT to allow its use outside an application environment.
- V3.0 The DEVTOOLS CONVERT/GENERATE command is now implemented. This is primary used to generate MMS scripts from application sources. Most converters are now implemented as code rather than TECO or TPU scripts, which has significantly improved their performance.

2.4.16.10 DELETE

- V3.1 The DEVTOOLS DELETE command now supports the /COMBINED qualifier. See HELP DEVTOOLS DELETE /COMBINED for details.
- V3.0 The DEVTOOLS DELETE command now supports the /TIDY and /SUBCONTEXT qualifiers. See HELP DEVTOOLS DELETE /TIDY for details.

2.4.16.11 DIFFERENCES/DATES

- V3.1 The DEVTOOLS DIFFERENCES/DATES command can now pass on the /COMBINED qualifier on the DEVTOOLS DELETE commands it generates.
- V3.0 The DEVTOOLS DIFFERENCES/DATES command can now pass on the /TIDY qualifier on the DEVTOOLS DELETE commands it generates.
- V2.6 The DEVTOOLS DIFFERENCES/DATES command now supports the /SINCE qualifier which specifies that only objects revised since the indicated time should be checked. This qualifier effectively applies only to the first parameter, and cannot be used with the /BOTH parameter.

2.4.16.12 RDU

- V2.6 The DEVTOOLS RDU[/KEEP] command invokes the TDMS RDU utility.

2.4.16.13 SET UTILITY/KEEP

- V2.6 Specifies that the indicated utility is to be operated in kept mode. Utilities supported include ADU, CDO and RDU.

2.4.16.14 SET UTILITY/NOKEEP

- V2.6 Specifies that the indicated utility is not to be operated in kept mode. If the utility is currently kept, it is topped.

2.4.16.15 SHOW CONTEXT

- V3.2 The DEVTOOLS SHOW CONTEXT command is a synonym for the UTLTOOLS SHOW COMMAND which displays the current context.

2.4.17 EDIT

- V2.5 If the default editor is LSEDIT, there is no sub-process editor and there is a DECwindows LSEDIT running in the appropriate application environment, group or user context, an EDIT command from a terminal window will push any file requested for editing across to the DECwindows LSEDIT window, and restore and focus it.

Note that an EDIT command without a file specification will continue to either start or attach to a sub-process style editor.

2.4.18 FDU

V2.4-3 The FDU command has been modified so that all terminal and DECwindows DECterm users will not need to set their terminal to VT100 format before using FDU. The command now invokes a command procedure which sets the terminal type to VT100 before invoking FDU and returns it to its prior type after FDU is finished.

Note also that EDITing a .FRM file now invokes FDU.

2.4.19 LADCP and LASTCP

V2.4-3 The LADCP and LASTCP commands are now defined for system managers.

2.4.20 LOGOUT

V2.4-3 A LOGOUT command has been implemented which checks to see if there are any sub-processes present below the current process, and if so issues a warning message rather than logging the user out. This prevents accidental deletion of a kept editor subprocess.

2.4.21 MAILBOX/DELETE

V3.0 The mailbox utility now provides a MAILBOX/DELETE command which deletes a mailbox.

2.4.22 REDUCE

V2.6-1 The REDUCE commands ignores many new errors codes. Use the following command to determine the complete set:

```
$ TYPE SWDEV_SFT_DIR:SWDEV_CVT_LOG_TO_ERR_EXCEPTIONS.TXT
```

V2.5 The REDUCE command removes general, success and selected informational messages from a log file. This is useful in tracking down actual errors in a large log file. In a future release of SysWorks this functionality will be placed in the DEVTOOLS utility.

2.4.23 SC

V3.2 The SC convenience command is now defined as UTLTOOLS SHOW CONTEXT.

V2.5-1 The SC (SHOW CONTEXT) convenience command no longer displays the saved DO command context since the DO command no longer supports memory and default qualifiers.

2.4.24 SPECIFIC

V2.4-3 Sets the application environment to the specific scope which will cause compilations and builds to occur in the developer private work and runtime directories.

2.4.25 SRCCTL

V2.6-1 The source control menu now includes an options to generate a report about the CMS library. This option is a front end to the DEVTOOLS CMS REPORT command.

V2.6 The source control menu now includes options to goto the change, testing and version control menus.

This menu is now available as **Manage** ⇒ **Sources** from the session manager menu bar.

2.4.26 TSTCTL

V2.6 The testing control menu now includes options to goto the change, source and version control menus.

This menu is now available as **Manage** ⇒ **Testing** from the session manager menu bar.

2.4.27 UTLTOOLS

This section briefly describes new and modified sub-sommands of the UTLTOOLS command.

2.4.27.1 DEFINE

V3.3 The UTLTOOLS DEFINE commands now support the /[NO]QUOTE qualifier which controls whther equivalence strings are quoted in generated command files.

2.4.27.2 General

V3.2 The UTLTOOLS utility is provided with SysWorks Base. Although it was provided is previous version of SysWorks, it was only used internally. In this release, it has become a generally available utility and will eventually be used to replace the DEFxxx commands.

The UTLTOOLS command is now documented. See online HELP or the *SysWorks Command Reference Manual* for details. The following list provides a command summary:

```
CREATE/NAME_TABLE
DEASSIGN
DEFINE
DEFINE/V1
DEFINE/VARIABLE
SET CONTEXT
SET FILE/HEADER
SHOW CONTEXT
SUBMIT
```

2.4.28 VSNCTL

V2.6 The version control menu has the same functionality as the previous change control menu. This menu now includes options to goto the change, source and testing control menus.

It is now available as **Manage** ⇒ **Versions** from the session manager menu bar.

2.4.28.1 TRIAL

V2.5 There is now a question in the TRIAL dialog allowing the check on CMS reservations to be overridden.

2.4.29 WHO

V3.2 A full show process utility named WHO is now provided. See HELP WHO for details.

2.5 Menus

This section briefly describes the new and modified menus provided in SysWorks.

2.5.1 CHGCTL, SRCCTL and TSTCTL Menu Help

V2.4-3 The CHGCTL, SRCCTL and TSTCTL menus now have menu specific help when the HELP selection is specified. It is the same as HELP @SWDOCHLP xxxCTL, where xxx is CHG, SRC or TST.

2.5.2 Environment Registration Menu

V2.6 Various tasks on the environment registration menu have had an option requesting the permitted environment scopes. By default, the environment scope is inherited from the environment type. If the inherited value is EITHER, an alternative scope value such as COMMON or SPECIFIC may be entered when adding or modifying the details about an environment.

2.6 Tasks

This section briefly describes SysWorks tasks which have been modified.

2.6.1 Merge maintenance into development

V2.4-2 The Merge maintenance into development task now asks whether to override the CMS library check like the VSNCTL TRIAL task.

2.7 Logical Names

This section briefly describes the logical names added or removed in various versions of SysWorks.

2.7.1 *appl*_ACMS_REPLACE

V2.4-3 The new *appl*_ACMS_REPLACE logical name in the application logical name table is used to indicate that ACMS sources (i.e. applications, menus, task groups and tasks) contain ADU utility commands such as create or replace, and that SysWorks should not issue these commands. By default, SysWorks issues these commands.

2.7.2 *appl*_APPLICATIONS

V2.4-3 The new *appl*_APPLICATIONS logical name in the application logical name table is used to place ACMS application definitions in a specific dictionary directory rather than CDD\$DEFAULT.

2.7.3 *appl_CMS_CLASS*

V2.6 The *appl_CMS_CLASS* logical names are no longer supported. These have been deprecated since V2.5 of SysWorks. The *appl_CMS_PATH* logical name should be used instead of *appl_CMS_CLASS*.

V2.4-3 The new *appl_CMS_CLASS* logical name in the application logical name table is used to control access to an applications common CMS library.

2.7.4 *appl_CMS_GENERATION*

V2.6 The *appl_CMS_GENERATION* logical names are no longer supported. These have been deprecated since V2.5 of SysWorks. The *appl_CMS_PATH* logical name should be used instead of *appl_CMS_GENERATION*.

V2.5 The new *appl_CMS_GENERATION* logical name in the application logical name table is used to control access to an applications common CMS library. For CMS commands which would logically use the /GENERATION qualifier (such as RESERVE), the DEVTOOLS utility will default to using this logical rather than the old *appl_CMS_CLASS* logical. If this logical is not present, the *appl_CMS_CLASS* logical is used with an appended "+", indicating that the latest generations of the class are to be used.

2.7.5 *appl_CMS_GROUP*

V2.4-3 The new *appl_CMS_GROUP* logical name in the application logical name table is used to control access to an applications common CMS library.

2.7.6 *appl_CMS_PATH*

V2.6 The *appl_CMS_PATH* logical name now supports a search list of generations. It must also be used in place of the *appl_CMS_CLASS* and *appl_CMS_GENERATION* logical name.

During the transition from V2.5-2 to V2.6 of SysWorks, it is possible to have both the *appl_CMS_CLASS* or *appl_CMS_GENERATION* logicals names defined at the same time as *appl_CMS_PATH*. This allows applications to change their LOGICALS.COM before V2.6 installation, and still work immediately afterwards.

To do this the following steps must be performed:

- Continue to define the *appl_CMS_CLASS* logical name without the trailing plus or the *appl_CMS_GENERATION* logical name with the trailing plus.
- For application environments (other than the common or APPL environment) which have their own CMS libraries, it is now necessary to have the appropriate class defined. For example, if the FIN application in the DEV environment has its own CMS library, create a FIN_DEV class in the library.

Insert a generation of each element into the class using a command similare to:

```
$ CMS INSERT GEN *.* /GEN=1+ FIN_DEV "Ready for V2.6"
```

- Create a class for each developer in the appropriate CMS library. These class names are of the for *appl_envr_user*. For example:

```
$ CMS CREATE CLASS FIN_DEV_JONES_AB "Ready for V2.6"
```

These classes should be left empty until after V2.6 is installed. There should be one of these classes for each developer or maintainer of each application environment.

- Modify the application's LOGICALS.COM to define both the *appl_CMS_GENERATION* and *appl_CMS_PATH* logical names. These should be further modified after SysWorks V2.6 installation to remove the definition of the *appl_CMS_GENERATION* logical name.
- Create developer specific sub-directories in the appropriate environments. For example, for the application FIN in environment DEV, the developer JONES_AB would currently have the following sub-directories:

— DISK_DEVnn:[FIN.RUN.JONES_AB]

— DISK_DEVnn:[FIN.WRK.JONES_AB]

The following new sub-directories need to be created:

— DISK_DEVnn:[FIN.LIB.JONES_AB]

— DISK_DEVnn:[FIN.SFT.JONES_AB]

Note that some sites may use their own directory codes in place of RUN, WRK, LIB and SFT.

2.7.7 *appl_CMS_VARIANT*

V2.4-3 The new *appl_CMS_VARIANT* logical name in the application logical name table is used to control access to an applications common CMS library.

2.7.8 *envr_DEVELOPER_STYLE*

V2.4-3 The meaning of the *envr_DEVELOPER_STYLE* logical name has been changed. The values of COMMON and SPECIFIC no indicate the default entry scope when an APPLICATION or ENVIRONMENT command is used.

2.7.9 *appl_GROUPS*

V2.4-3 The new *appl_GROUPS* logical name in the application logical name table is used to place ACMS task group definitions in a specific dictionary directory rather than CDD\$DEFAULT.

2.7.10 *appl_MENUS*

V2.4-3 The new *appl_MENUS* logical name in the application logical name table is used to place ACMS menu definitions in a specific dictionary directory rather than CDD\$DEFAULT.

2.7.11 *appl_MSGHLP_LIB*

V2.4-3 The new *appl_MSGHLP_LIB* logical name in the application logical name table is used to identify the HELP/MESSAGE data file used by the application (if any).

2.7.12 *appl_TASKS*

V2.4-3 The new *appl_TASKS* logical name in the application logical name table is used to place ACMS task definitions in a specific dictionary directory rather than CDD\$DEFAULT.

2.7.13 *appl_TDMS_REPLACE*

V2.4-3 The new *appl_TDMS_REPLACE* logical name in the application logical name table is used to indicate that TDMS sources (i.e. request and request library definitions) contain RDU utility commands such as create or replace, and that SysWorks should not issue these commands. By default, SysWorks issues these commands.

2.7.14 *SWDEV_DEVELOPER_STYLE*

V2.4-3 This logical name defines the default entry scope used when a developer enters an application environment with the *APPLICATION* or *ENVIRONMENT* commands. It is overridden by the *appl_DEVELOPER_STYLE* logical name when that is present (eg. defined in an application's *LOGICAL.COM* procedure).

2.7.15 *SWRK_ENVR_envr*

V2.5-1 The *SWRK_ENVR_WRK_envr* and *SWRK_ENVR_VACL_envr* logical names in the *LNМ_SWRK_DATABASE* logical name table have been superseded by the *SWRK_ENVR_envr* logical names which contain all the information about an environment. For public level installations of SysWorks, these logical names are usually defined in the site specific pre startup procedure *SWRK_LCL_DIR:site_PRE_STARTUP.COM* which should be changed to reflect the new logical names.

The following DCL commands illustrates how one of these logical names might be defined:

```
$ DEFDAT := DEFINE/TABLE=LNМ_swrk_DATABASE  
$ DEFDAT SWRK_ENVR_DEV "SCOP=SPECIFIC\WORK=1\VACL=0\VRNT=0\VRSN=0"
```

Note that the equivalence must be in uppercase and inside quotes. The five properties have the following meanings:

Property	Value	Meaning
SCOP	COMMON	Only the common scope is supported
	SPECIFIC	Both command and specific scopes are supported
WORK	0	No work directories are supported
	1	Work directories are supported
VACL	0	CMS variants should not be used
	1	CMS variants should be used
VRNT	0	No multi-variant support
	1	Multi-variant support
VRSN	0	No multi-version support
	1	Multi-version support

The following are the standard defaults:

Environment	Properties
APPL	SCOP=COMMON\WORK=0\VACL=0\VRNT=0\VRSN=0
DEV	SCOP=SPECIFIC\WORK=1\VACL=0\VRNT=1\VRSN=0
DTST	SCOP=COMMON\WORK=1\VACL=0\VRNT=1\VRSN=0

Environment	Properties
FDEV	SCOP=SPECIFIC\WORK=1\VACL=0\VRNT=1\VRSN=0
MNT	SCOP=SPECIFIC\WORK=1\VACL=1\VRNT=0\VRSN=1
MTST	SCOP=COMMON\WORK=1\VACL=1\VRNT=0\VRSN=1
PROD	SCOP=COMMON\WORK=0\VACL=0\VRNT=0\VRSN=0
TRNG	SCOP=COMMON\WORK=0\VACL=0\VRNT=0\VRSN=0

Note that if ACMS is present on a computer node, the scope property is overridden to be common, since ACMS cannot support the SPECIFIC scope context.

2.7.16 SWRK_ENVR_VACL_enuv

- V2.5-1 The SWRK_ENVR_VACL_enuv logical name has been superceded by the SWRK_ENVR_enuv logical name.
- V2.4-3 Revised SWRK_ENVR_VACL_enuv logical name (in the LNM_SWRK_DATABASE logical name table for non-private installation levels) is used to specify CMS variant class control flags. If this logical has a true value, the generations in the application environments CMS class must be variants.

2.7.17 SWRK_ENVR_WRK_enuv

- V2.5-1 The SWRK_ENVR_WRK_enuv logical name has been superceded by the SWRK_ENVR_enuv logical name.
- V2.5 New SWRK_ENVR_WRK_enuv logical name to control which environments should have work areas. This logical name should be placed in the LNM_swrk_DATABASE logical name table, which will be system wide when SysWorks is installed at or above the public installation level. For sites which install SysWorks at or below the public level, this logical name should be defined in the site specific SysWorks pre startup command procedure.

2.7.18 SWRK_UTILITY_LOGICALS_SCOPE

- V2.5-1 This logical name has an equivalence of SYSTEM or USER and controls where the utility logical names supported in SysWorks are defined.

2.7.19 SQL\$USER and LNK\$LIBRARY_nn

- V2.6-1 The SQL\$USER logical name is no longer defined as a LNK\$LIBRARY_nn logical name. This is to allow the use of multi-variant DEC Rdb and part of a trend away from using LNK\$LIBRARY_nn logical names. In order to use the SQL\$USER object library, each the Linker option file for each image which uses SQL should include a line such as:

```
sql$user/library
```

Regarding the trend away from LNK\$LIBRARY_nn logical names, application Linker options files should include a line similar to the above for each library they need to link against. It is recommended that each library be represented by a logical name.

The preferred SysWorks format for such logical names include the following:

Logical Name	Equivalence	Usage
<i>appl_HELP_LIB</i>	<i>appl_DOC_DIR:applHLP.HLB</i>	Application help library
<i>appl_IMAGE_LIB</i>	<i>appl_SFT_DIR:applIMG.OLB</i>	Application shareable image library
<i>appl_OBJECT_LIB</i>	<i>appl_LIB_DIR:applOBJ.OLB</i>	Application object library
<i>appl_SYMBOL_LIB</i>	<i>appl_LIB_DIR:applSYM.OLB</i>	Application symbol library (if separate from the object library)
<i>appl_TEXT_LIB</i> or <i>appl_COPY_LIB</i>	<i>appl_LIB_DIR:applTXT.TLB</i>	Application Cobol, DECforms and/or Fortran include library

2.8 Routines

The following routines were provided as part of SysWorks as a public interface into the SWRKSHR shareable image.

2.8.1 SWRK_BINARY_SEARCH

V3.2 This routine searches an ordered table of keywords using a binary algorithm.

2.8.2 SWRK_CLOSE_OUTPUT

V2.6-1 This routine closes the SysWorks standard output stream.

2.8.3 SWRK_DISPLAY_HELP

V3.2 This routine displays OpenVMS help.

2.8.4 SWRK_DUMP_IMAGE_INFO

V2.6 This routine dumps an image list and frame analysis. It takes one argument which is the address of an outout callback routine. If this routine is not present, the output is directed to SYS\$OUTPUT.

2.8.5 SWRK_ESTABLISH

V2.6 This routine established the SysWorks exception handler. Its behaviour changes depending on the execution mode of the caller, or whether the caller was within the main image or an executable image. If the debugger is present, no exception handler is established, since it is assumed that the debuggers exception handler will suffice.

2.8.6 SWRK_EXCEPTION_HANDLER

V2.6-1 This routine now detects PCA\$COLLECTOR in addition to DEBUG for the purpose of not handling most exceptions. This is so that these images can trap exceptions.

V2.6 This routine is the actual exception handler which is established by SWRK_ESTABLISH. It takes the arguments specified in the OpenVMS Introduction to System Services manual.

2.8.7 SWRK_EXTENDED_FILE_SEARCH

V3.3 This routine has been enhanced to support the /SELECT qualifier. As a consequence, all SysWorks commands which use SWRK_EXTENDED_FILE_SEARCH now support the /SELECT qualifier. All such commands have had their documentation updated to reflect this.

2.8.8 SWRK_HANDLE_IMAGE_VECTOR

V2.6-1 This JSB routine is used to handle SysWorks format shareable image vectors.

2.8.9 SWRK_HANDLE_QUALIFIERS

V3.3 This routine has been enhanced to allow a leading double ampersand to be interpreted as a single ampersand rather than as a symbol substitution operator.

V3.2 This routine has been enhanced to allow a leading ampersand to be interpreted as symbol substitution operator. The command parameter or qualifier value will be that of the symbol name following the ampersand rather than the value initially passed by the used. If the symbol named after the ampersand is not defined, a blank value results.

2.8.10 SWRK_OPEN_OUTPUT

V2.6-1 This routine opens the SysWorks standard output stream.

2.8.11 SWRK_PUT_OUTPUT

V2.6-1 This routine writes a line to the SysWorks standard output stream.

2.9 Macros

This section briefly describes the new and changed Macros provided in SysWorks.

2.9.1 FIXUP

V2.6-1 The FIXUP macro replaces the previous UPDATE macro. It is used to fixup address information in a data structure explicitly at runtime rather than at image activation time.

2.9.2 UPDATE

V2.6-1 The new UPDATE macro performs an update on a file and is thus now related to the CREATE, GET, OPEN and PUT macros.

2.10 MMS Generators

This section briefly describes the new or updated MMS generators provided in SysWorks.

2.10.1 Improvements to All MMS Generators

V3.2 All MMS generators have now been enhanced to allow a logical name to be used for a file based dependency. This feature was first introduced for Cobol, and is now used throughout all MMS generators which make dependencies on files. There are a number of important side effects of this change. For example, in conjunction with the *appl_EXPLICIT_LINK_FLG* logical name, it is now possible to have more than one object library within an application.

2.10.2 ACMS

V3.1 All ACMS sources now generate dependencies for `%INCLUDE` statements. These dependencies now work for both `ADU REPLACE` and `ADU BUILD` commands. Note that only complete statements can be present within an include source at this time. Examples of such statements include:

```
forms are ... ;
procedures are .... ;
tasks are ... ;
workspaces are ... ;
```

2.10.3 ACMS Task Group

V3.1 Many improvement and reorganizations have been made to the ACMS Task Group MMS script generator. The aim of these has been to increase the level of automation and reduce the level of application specific MMS script used with these.

The following changes have been made:

- Targets for an ACMS Task Group build are now generated. These are designed to combine dependencies required to build the `.TDB` file and the various server images associated with the ACMS Application.
- Dependencies are now generated to automatically generate DECforms Linker options or options include files. These should no longer be present in the application MMS scripts associated with the `RULES` or `DESCRIP` phases.
- When generating dependencies relating to DECforms Linker options files, SysWorks checks to see if an associated Linker options file exists in the work directory. If so it assumes that this source includes a `%APPEND` command for an associated Linker options include file and generates that. If the source is not found in the work directory, SysWorks generates a complete Linker options file which is architecture specific.

For example if the ACMS Task Group definition includes a reference to `FIN_SFT_DIR:FINFRM.EXE` and there exists a `FIN_WRK_DIR:FINFRM.OPT` source, a `FIN_AIL_DIR:FINSHR.OPT_INC` is generated which contains the list of forms which are placed in the shareable image. If `FIN_WRK_DIR:FINFRM.OPT` doesn't exist, a `FIN_LIB_DIR:FINFRM.OPT` is generated with options appropriate for the architecture on which it is being generated.

- Images within the form list can now be a logical name. SysWorks performs one level of logical name translation to get the actual image file name.

For a complete description of all the dependencies generated for an ACMS Task Group source and the assumptions made in order to generate them, please refer to the *SysWorks Application Development Guide*.

2.10.4 ADA

V2.6 There is a new MMS generator variant for ADA sources with a file type of `.ADA`.

2.10.5 Basic

V2.6 There is a new MMS generator variant for Basic sources with a file type of `.BAS`.

2.10.6 Bookreader

- V2.6** There is a new MMS generator variant for Bookreader bookshelf sources with a file type of `.DECW$BOOKSHELF_SRC`.

2.10.7 C

- V3.2** A new logical name of the form `appl_EXPLICIT_LINK_FLG` now control whether the MMS generator for C creates link time dependencies in addition to compile time dependencies.

The default value of NO causes the generator to behave as it did previously i.e. it uses `.TAG_EP` dependencies to correctly build images.

A value of YES stops this behaviour. Under these circumstances, the best method of linking to ensure correct builds is to explicitly list each module in a Linker options file, either as an object file or an included modules from an object library. Benefits of this approach include:

- C sources and/or include modules don't need to use the `extern` statement to cause the `.TAG_EP` dependency.
- Old style function headers with argument names only within the function declaration parenthesis and argument data types declared after the header can be supported. Note that this style of function header should be discouraged as it neither conforms to standards nor can it be used in a prototype.

These benefits allow SysWorks to be used with legacy or external code without the need for substantial modifications.

2.10.8 C

- V3.1** The MMS generator for C now generates dependencies for include files whose names do not start with the application code. In order to distinguish between application and system include files, a search for the file is made in the application work directory. If it is present, a dependency is made on it. If it is not present, it is assumed to be a system include file and no dependency is generated (i.e. the previous behaviour).

Note that no search for the file is made if its name starts with the application code (i.e. same as previous behaviour). Thus it is more efficient to continue the practise of using the application code as a file name prefix.

This enhanced behaviour is primarily intended to support legacy or imported code rather than relaxing the recommendation to prefix all sources with the application code.

2.10.9 C++

- V2.6** There is a new MMS generator variant for C++ sources with a file types of `.CXX`, `.HXX` and `.IXX`.

2.10.10 Cobol

V3.0 The Cobol MMS generator has been enhanced to allow a logical name to be used for a copy library. SysWorks performs one level of logical name translation in order to determine the actual location of the copy library for dependency purposes.

V2.6 The Cobol MMS generator has been enhanced to allow for transitive dependencies in copy statements. For example, if a copy file includes a call statement, the MMS scripts will ensure that the called routine will be compiled before any image using the Cobol program is linked.

There is a new MMS generator for Cobol and DECforms copy files and libraries. By default, the file type for these copy files is .TXT. The alternative file type is .LIB.

2.10.11 DECdocument

V3.2 The MMS generator for DECdocument now detects the presence of <CONTENTS_FILE> and <INDEX_FILE> tags in order to control the addition of /CONTENTS and /INDEX qualifiers to the DOCUMENT command. This eliminates problems if a document which SysWorks previously expected to have a contents section and an index not having either or both. Note that this feature only applies to .SDML sources, not .SDML_INC include files. Thus, the <CONTENTS_FILE> and <INDEX_FILE> tags must be placed in the main .SDML source.

At this time, the COMPILE command continues to assume that certain document types have both contents and index.

V2.6-1 The MMS generator for DECdocument now uses the SysWorks design files and places them in the dependency list for each document.

V2.6 There is a new MMS generator variant for DECdocument sources with a file type of .SDML. Table 2–9 lists the targets for which dependencies will be created based on the end of the filename.

Table 2–9 Targets from File Name suffices

File Name Suffix	Target File Types
_CVR	.PS, .TXT
_GDE	.DECW\$BOOK, .PS, .TXT
_HLP	.HLP, <i>app</i> HLP.HLB
_REF	.DECW\$BOOK, .PS

All the targets will be placed in the applications documentation directory.

2.10.12 DECdocument Graphics

V3.0 There is a new MMS generator for DECdocument graphics objects with a file type of .GRA.

2.10.13 DECforms

V3.1 The DECforms MMS generator no longer defines dependencies on fields within records in response to the `field is` clause. The dependency is now on the record structure as a whole. This forces a recompilation whenever the record changes rather than just the field, which produces a more consistent build.

2.10.14 DECforms

- V2.6** The DECforms MMS generator has been enhanced to allow for transitive dependencies in copy statements. For example, if a copy file includes a call statement, the MMS scripts will ensure that the called routine will be compiled before any image using the DECforms form is linked.

2.10.15 Linker

- V3.1** The `%NODEBUG` command indicates that this image is always linked `/NODEBUG/NOTRACEBACK` even when the `/DEBUG` or `/TRACEBACK` qualifier is used with the `BUILD` or `COMPILE` command.

- V3.0** The MMS generator for Linker options files now interprets some commands within comments to control the way in which an image is linked (similar to the `%APPEND` ability within DCL command procedure sources).

The `%APPEND` command indicates that an additional Linker options file should be included in the `LINK` command. This is useful when different syntax or options are used for Alpha and VAX images. For example, `FIN_MAIN-ALPHA.OPT` and `FIN_MAIN-VAX.OPT` might each have a set of separate options, but also include a `%APPEND FIN_MAIN_COMMON.OPT_INC` command which causes the MMS script to contain a syntax similar to:

```
$(link) FIN_LIB_DIR:FIN_MAIN.OPT/options, -  
      FIN_WRK_DIR:FIN_MAIN_COMMON.OPT_INC/options
```

Note that the Linker main source comes from the library directory since the original source has an architecture specific suffix which causes an unsuffixed copy to be made in the library directory.

- V2.6-1** There is a new feature in the MMS generator for Linker options files. If the filename of the image ends in the suffix `TOOLS`, the image is always linked `/NODEBUG/NOTRACEBACK`.

2.10.16 Macro

- V2.6** The Macro MMS generator now differentiates between Macro sources which generate code and those which purely define macros and symbols. If the file name of a Macro source ends with `_MAC`, it is assumed to only contain macros. Each macro within such a source which contains a `$define` directive, is assumed to define symbols. The resultant MMS script reflects these attributes.

2.10.17 Pascal

- V2.6** There is a new MMS generator variant for Pascal sources with a file type of `.PAS`.

2.10.18 Linker

- V2.5-2** The MMS generator for Linker options files has been enhanced.

2.11 Command Procedures

This section briefly describes the command procedures with modified actions in SysWorks.

2.11.1 SYLOGIN.COM

V2.6-1 The SysWorks SYLOGIN.COM no longer executes a REPLY/ENABLE for processes which are interactive, have a terminal (excluding DECwindows terminals) as their input stream as possess the OPER or SETPRV privilege as a default or authorised privilege.

2.12 DECwindows

This section briefly describes the modified DECwindows features provided by SysWorks.

2.12.1 DECwindows Utilities Pulldown Menu

V2.6 The DECwindows Utilities pulldown menu will now include POSIX (POSIX style), POSIX (OpenVMS style) and/or VUIT entries where appropriate.

2.12.2 SysWorks Manage Pulldown Menu

V2.6 The SysWorks Manage pulldown menu now has fewer items on it. Many items now display a menu consisting only of other menus.

2.12.3 CPU Load and Eyes

V2.5-2 The CPU Load and Eyes items have been moved from the Backgrounds menu to the Utilities menu.

2.12.4 DECwindows/Motif

V2.5 At this time only V1.0 of Motif should be used with SysWorks. There have been significant problems when advanced SysWorks V2.5 and DECset V11 menus and item lists have been used with Motif V1.2.

2.12.5 .DOC and .DOC_STYLE File Types

V2.5 The .DOC and .DOC_STYLE mouse button three file type handling has been extended to support most actions that might be required from such a file type.

2.13 Miscellaneous

This section briefly describes miscellaneous enhancements provided in SysWorks.

2.13.1 SWDEV_VERY_OLD_FILE.TXT

V2.6-1 A new file has been provided with SysWorks which has very old creation and modification dates of 01-Jan-1900. It is used by SysWorks as a universal MMS source.

2.13.2 Message Symbol Code Prefix Change

V3.0 All SysWorks message symbols now start with a prefix of SWRK__.

2.13.3 Display of messages in hook files

V2.5 In previous versions of SysWorks, the various user and application hook files (eg. ENTER.COM, EXIT.COM and HOME.COM) displayed their own 'Executing ...' messages. These are now displayed by the various SysWorks command such as APPLICATION and HOME. In a future release of SysWorks this display will be controlled by a context variable (i.e symbol, logical name or SysWorks variable). For the time being, they will always be displayed.

As a result, users and developers should remove these messages from the hook procedures.

This chapter briefly describes the problems with SysWorks which have been fixed.

3.1 Commands

This section briefly describes problems with commands which have been fixed in SysWorks.

3.1.1 Many Commands

Many commands use the `SWRK_EXTENDED_FILE_SEARCH` routine which has had the handling of `/BEFORE` and `/SINCE` fixed. Therefore all SysWorks™ commands which use `/BEFORE` and `/SINCE` have consequentially been fixed.

3.1.2 ACCEPT

V3.2 The `ACCEPT` command failed when the cursor was not at the end of the input when a terminator was pressed.

This has been fixed.

3.1.3 CONTEXT

V2.6-1 The `CONTEXT` command will attempt to create the appropriate logical name table when moving to a new context and the logical name table is not present. In previous versions of SysWorks if the user had privileges, the logical name table would be created in the system logical name table, but without the ACL necessary for other users of the context to have access.

This has been fixed by always creating such a logical name table in the process logical name table.

In a future version of SysWorks, all valid users of a context will be able to create a system wide logical name table with the appropriate ACL. This will improve login times considerably and give public installation sites the same logical name table abilities as system and turnkey installations.

3.1.4 DEVTOOLS

This section briefly describes problems with the `DEVTOOLS` utility which have been fixed.

3.1.4.1 CMS CREATE ELEMENT

V2.6-1 The DEVTOOLS CMS CREATE ELEMENT command did not behave correctly when the element existed in the CMS library without a generation being a member of the environment or versions class.

This has been fixed.

3.1.4.2 CMS DELETE ELEMENT/[NO]CONFIRM

V2.6 The /[NO]CONFIRM qualifier is now correctly supported by the DEVTOOLS CMS DELETE GENERATION command. This makes the output from a DEVTOOLS DIFFERENCES/DATES (i.e. COMPARE) command work properly.

3.1.4.3 CMS FETCH

V2.6-1 The DEVTOOLS CMS RESERVE command did not behave correctly when attempting to reserve a generation which needed promotion.

This has been fixed.

3.1.4.4 CMS INSERT GENERATION

V3.3 The DEVTOOLS CMS INSERT GENERATION command didn't insert any generations in a list when any generation wasn't found. The command now correctly reports errors but still inserts generations where possible.

3.1.4.5 CMS REPLACE

V2.6-1 The DEVTOOLS CMS REPLACE command did not insert the new generation into the class specified by *appl_CMS_PATH* when use of the latest generation was indicated for the path (i.e. the path did not end with a plus sign).

This has been fixed.

V2.5-1 The DEVTOOLS CMS REPLACE command failed to update the application environment class when the translation of the *appl_CMS_GENERATION* logical name did not include a trailing plus which indicated that generations from the indicated class should be used rather than the latest generations from the class. This has now been fixed.

3.1.4.6 CMS RESERVE

V3.3 The DEVTOOLS RESERVE command wouldn't reserve a generation which needed promotion along its main line of descent when that generation didn't have a variant and the default for the reservation context did. A generation could be reserved in such a situation because a subsequent replace would have created a variant generation rather than a conflicting newer generation in the same line of descent.

The command will now reserve a generation in these circumstances.

3.1.4.7 CONVERT/GENERATE

V3.2 The DEVTOOLS CONVERT/GENERATE command caused an image failure when attempting to open a file for which the user did not have read access.

This has been fixed. An appropriate error message is now reported.

3.1.4.8 DIFFERENCES/DATES

- V2.6-1** The DEVTOOLS DIFFERENCES/DATES command has problems when the /MERGE qualifier was used and there was no generation in the target class. This has been fixed.

3.1.4.9 DIFFERENCES/DATES

- V2.5-1** Some combinations of comparing a directory with a CMS library and comparing two CMS libraries did not function properly. These have been fixed.

3.1.5 CHGCTL

- V2.5-1** Various CHGCTL sub-commands performed actions in application environment contexts. When these commands were performed the default directory was the users runtime directory rather than the users work directory. This has now been fixed.

3.2 Menus

- V2.6-1** When a SysWorks menu has more entries than can be displayed on the screen at one time, the menu is paged and supports the **Prev** and **Next** keys for navigation. In earlier versions of SysWorks, this feature did not work properly. This has been fixed.

3.3 Routines

The following routines have had problems fixed.

3.3.1 SWRK_HANDLE_QUALIFIERS

- V3.3** This routine didn't provide or handle a default date value. This has now been fixed.

3.3.2 SWRK_MOVE_FILE

- V3.3** This routine didn't delete the input file when necessary if there wasn't an explicit version. Also, when a copy was required, the output file parse could fail on the second file. These problems have now been fixed.

3.4 Tasks

This section briefly describes problems with tasks which have been fixed in SysWorks.

3.4.1 Rename a disk volume

- V2.6** The Rename a disk volume task failed when attempting to rename a non clusterwide disk volume from one of the computer nodes on which it is mounted when that computer node was not a cluster master computer node.

This has been fixed by allowing any computer node on which the disk is mounted to rename the disk volume.

3.4.2 Merge maintenance into development

V2.5-2 Various problems with the Merge maintenance into development task have been fixed.

3.5 MMS Generators

This section briefly describes problems with the SysWorks MMS generators which have been **fixed**.

3.5.1 ACMS Menu Definition

V3.1 The MMS generator now correctly parses menu *is cdd_path* where the *cdd_path* starts with a non symbol constituent character.

3.5.2 CDD/Repository

Various MMS generators (such as ACMS) which used the CDD had problems when the CDD path started with a non symbolic character. These generators should now generate the correct syntax.

3.5.3 CDD/Repository

V3.1 The MMS generator now correctly parses computed by clauses on field definitions which include arithmetic and string operators, numeric or string literals and parentheses.

3.5.4 Cobol

V2.6-1 The MMS generator now accepts single quotes as string bound in the same ways as double quotes.

The MMS generator now make dependencies on external (i.e. in another application) copy libraries use `SWDEV_SFT_DIR:SWDEV_VERY_OLD_FILE.TXT` as the source for the item. This file has a date which makes it older than any normal source so that no attempt is made to update the external copy library.

It also now accepts the `copy ... of ...` syntax as well as `copy ... in`

3.5.5 DECforms

V3.2 The MMS generator now correctly handles copy statements which have the `from` or `in` clauses on a separate line to the initial `copy` statement.

3.5.6 DECforms

V2.6-1 The MMS generator now accepts single quotes as string bound in the same ways as double quotes.

The MMS generator now make dependencies on external copy libraries in the same way as the Cobol MMS generator (see above for details).

3.5.7 ACMS Application

V2.6 If the server control attributes contained clauses before the server *name* in ... clause, the generator did not generate the correct dependency.

This has been fixed.

3.5.8 C

V2.6 If the main() routine had a prototype, the generator created a non-main style of dependencies.

This has been fixed.

3.5.9 Cobol

V2.6 If a call statement contained a space in the called routine name, this space appeared in the generated MMS script. This problem has been fixed.

If there were any complete strings (i.e. test between two double quotes) before the call statement, the generator did not generate a dependency. This problem has been fixed.

3.5.10 TDMS Request Library

V2.6 The use of .TAG_EP and .TAG_CDD was incorrect. This was left over from when the generated MMS script attempted to distinguish between when the request library source needed to be replaced versus when it only needed to be rebuilt. Since V2.5 of SysWorks, this distinction has not been made i.e. when either the request library source or the source of any its requests was changed, the request library is both replaced and built. This problem has been fixed.

3.5.11 CDO

V2.5-2 The MMS generator for CDO source file types has been enhanced to support the use of fields outside the application.

3.5.12 DECforms

V2.5-1 The DECforms field is syntax was incorrectly interpreted by the MMS generator.

This has now been fixed.

3.5.13 TDMS Request Library

V2.5-1

3.5.14 C

V2.4-3 The MMS generator for the C, CXX, H, HXX, RC and SC source file types had several problems with data enclosed in quotes appearing like C code and also problems with outer level function recognition.

These have been fixed.

3.6 Command Procedures

This section briefly describes the command procedures fixed in SysWorks.

3.6.1 SWDEV_CVT_RDB_TO.COM

V2.6-1 This command procedure has a number of problems related to extended syntax in databases created by RDO rather than SQL.

These have now been fixed.

3.7 Miscellaneous

This section briefly describes miscellaneous problems which have been fixed in SysWorks.

3.7.1 REPLY/ENABLE

V2.6-1 Earlier versions of SysWorks enabled OPCOM messages (i.e a REPLY/ENABLE command was issued) for users with the OPERATOR privilege. This feature is disabled for public installations.

3.7.2 LSEDIT Buffer Initialization

V2.4-3 There was a problem when EVE or LSEDIT were started as a sub-process for the first time in a job which resulted in the \$MAIN buffer being displayed rather than a specified file.

This has been fixed.

3.7.3 LSEDIT Initialization Files

V2.4-3 There was a problem when LSEDIT was started under DECwindows that resulted in the temporary initialization file remaining after LSEDIT had been initialized, and sometimes even after it had finished.

This has been fixed.

Known Limitations or Problems

This chapter briefly describes the known limitations and problems with SysWorks.

4.1 Entry Points in MACRO-32

Entry points used for dependency analysis in the MMS generator require the `.ENTRY` or `PROCEDURE` (supported in SysWorks SWDEVMAC.MLB macro library) syntax.

4.2 Entry Points in C

External references in C, CXX, H, HXX, RC and SC sources must be declared with the `EXTERN` keyword.

4.3 CVTMMS various

There are some inconsistencies in how various source types (eg. `.COB` and `.IFDL`) handle copies or includes from the CDD. These will be made fully consistent in a future release of SysWorks.

The following guidelines on the future standard treatment should help:

- Dictionary items starting with `appl_RECORDS`. will always be treated as a local (i.e. application) record, with dependencies built around a `.TAG_CDD` file in the application library directory.
- Dictionary items starting with `appl_FIELDS`. will always be treated as a local field with `.TAG_CDD` dependencies.
- Dictionary items starting with the application code and ending with `REC` will be treated as local records with `.TAG_CDD` dependencies.
- Dictionary items without any periods and ending with `REC` will be treated as a local record with `.TAG_CDD` dependencies.
- Dictionary items without any periods will be treated as a local field with `.TAG_CDD` dependencies.
- Other dictionary items will be treated as external items using the MMS `~^` (i.e. uparrow) dependencies.

Note that the dictionary item will be checked in the order indicated in the above list.

C

CHGCTL, 3-3

Command Procedures

SWDEV_CVT_RDB_TO.COM, 3-5

SYLOGIN.COM, 2-34

SYS\$LIBRARY:DECRDB_SETVER, 2-6

SYS\$LIBRARY:RDBVMS_SETVER, 2-6

Commands

ACCEPT, 2-14, 3-1

BUILD, 2-14, 2-15, 2-16, 2-33

CHANGE, 2-16

CHGCTL, 2-16, 2-23

COMMON, 2-16

COMPARE, 2-16

COMPILE, 2-16, 2-33

CONTEXT, 2-16, 3-1

APPLICATION, 2-16

ENVIRONMENT, 2-16

CVTMMS, 2-17

CVTRDB, 2-17

DEVTOOLS, 2-18, 3-1

ADU, 2-19

ATTACH, 2-19

CDO, 2-19

CMS CREATE ELEMENT, 3-2

CMS DELETE ELEMENT, 3-2

CMS FETCH, 2-19

CMS INSERT GENERATION, 3-2

CMS PROMOTE, 2-19

CMS REPLACE, 3-2

CMS REPORT, 2-19

CMS RESERVE, 2-19, 3-2

CMS SHOW GENERATION, 2-19

CONVERT/GENERATE, 2-20, 3-2

DELETE, 2-20

DIFFERENCES/DATES, 2-20, 3-3

RDU, 2-20

SET UTILITY/KEEP, 2-20

SET UTILITY/NOKEEP, 2-20

SHOW

CONTEXT, 2-20

DIRTOOLS, 2-17

Commands (cont'd)

DO, 2-18

EDIT, 2-20

FDU, 2-21

LADCP, 2-21

LASTCP, 2-21

LOGOUT, 2-21

MAILBOX/DELETE, 2-21

RBAS, 2-17

RCOB, 2-17

RDBCOBOL, 2-17

RDMLC, 2-17

RDMLPAS, 2-17

REDUCE, 2-21

REPLY/ENABLE, 3-6

RFOR, 2-17

SADA, 2-17

SBAS, 2-17

SC, 2-21

SCC, 2-17

SCOB, 2-17

SFOR, 2-17

SPAS, 2-17

SPECIFIC, 2-21

SPLI, 2-17

SRCCTL, 2-22, 2-23

TSTCTL, 2-22, 2-23

UTLTOOLS, 2-22

DEFINE, 2-22

VSNCTL, 2-22

TRIAL, 2-23

WHO, 2-23

D

DECdocument, 2-13

DEC Rdb

Precompier, 2-17

DEVTOOLS

CMS REPORT, 2-22

Directories

Test, 2-10

DIRTOOLS

DOCUMENTATION/REPLICATE, 2-2, 2-17

I

Images

DEBUG, 2-28
PCA\$COLLECTOR, 2-28

L

Layered Products

ObjectBroker, 2-13

Logical names

appl_ACMS_REPLACE, 2-23
appl_APPLICATIONS, 2-23
appl_CMS_CLASS, 2-24
appl_CMS_GENERATION, 2-24
appl_CMS_GROUP, 2-24
appl_CMS_PATH, 2-24
appl_CMS_VARIANT, 2-25
appl_GROUPS, 2-25
appl_MENUS, 2-25
appl_MSGHLP_LIB, 2-25
appl_TASKS, 2-25
appl_TDMS_REPLACE, 2-26
envr_DEVELOPER_STYLE, 2-25
LNK\$LIBRARY_{nn}, 2-27
MSGHLP\$LIBRARY, 2-4
SQL\$USER, 2-27
SWDEV_DEVELOPER_STYLE, 2-26
SWRK_ENVR_VACL_{envr}, 2-27
SWRK_ENVR_WRK_{envr}, 2-27

Logical Names

SWRK_ENVR_{envr}, 2-26
SWRK_TST_DIR, 2-10
SWRK_UTILITY_LOGICALS_SCOPE, 2-27

LSEDIT, 3-6

M

Macros, 2-29

FIXUP, 2-29
UPDATE, 2-29

Meta object Model, 2-8

MMS Generator

ACMS
Application, 3-4
ACMS Menu Definition, 3-4
C, 2-31, 3-5
CDD/Repository, 3-4
Cobol, 3-4
DECforms, 3-4

MMS Generators, 2-29

ACMS, 2-30
ACMS Task Group, 2-30
ADA, 2-30
All, 2-29
Basic, 2-30
Bookreader, 2-31
C, 2-31, 3-5
C++, 2-31
CDO, 3-5
Cobol, 2-32
COBOL, 3-5
Copy Files and Libraries, 2-32
DECdocument, 2-32
DECdocument Graphics, 2-32
DECforms, 2-32, 2-33, 3-5
Linker, 2-33
Macro, 2-33
Pascal, 2-33
TDMS
Request Library, 3-5

MMS scripts

appl_DOCUMENTATION.MMS, 2-15
DESCRIP.MMS, 2-15

P

Privileges

OPERATOR, 3-6

Q

Qualifiers

/BEFORE, 3-1
/SINCE, 3-1

R

Routines

SWRK_BINARY_SEARCH, 2-28
SWRK_CLOSE_OUTPUT, 2-28
SWRK_DISPLAY_HELP, 2-28
SWRK_DUMP_IMAGE_INFO, 2-28
SWRK_ESTABLISH, 2-3, 2-28
SWRK_EXCEPTION_HANDLER, 2-28
SWRK_EXTENDED_FILE_SEARCH, 2-29
SWRK_HANDLE_IMAGE_VECTOR, 2-29
SWRK_HANDLE_QUALIFIERS, 2-29, 3-3
SWRK_MOVE_FILE, 3-3
SWRK_OPEN_OUTPUT, 2-29
SWRK_PUT_OUTPUT, 2-29

T

Tasks

- Merge maintenance into development, 2-23, 3-4
- Rename a disk volume, 3-3
- Test Directory, 2-10

U

UTLTOOLS

- CREATE/NAME_TABLE, 2-22
- DEASSIGN, 2-22
- DEFINE, 2-22
- DEFINE/V1, 2-22
- DEFINE/VARIABLE, 2-22
- SET
 - CONTEXT, 2-22
- SET FILE/HEADER, 2-22
- SHOW
 - CONTEXT, 2-22
- SUBMIT, 2-22

