SysWorks Introduction

A Tutorial on Using SysWorks

Simon L. Jackson
SysWorks Development Team

Part I

Concepts

Introduction

SysWorks provides:

- **Enhanced DECwindows/Motif integration for users and application developers.**
- **Application development tools including building utilities and version management.**
- **System management tools including user management, configuration management and system startup and shutdown facilities.**

Installation Level

SysWorks may be installed at the following levels:

- **Public**

  Access to functions via users personal username.

- **System**

  Access to non-privileged functions via personal username. Access to privileged functions via servers.

- **Turnkey**

  Access to functions the same as for a system installation. Replaces all site-specific OpenVMS command procedures so that SysWorks controls all system management functions.

Documentation

SysWorks provides the integrated documentation in the following formats:

- **Bookreader**
- **Printable PostScript**
- **Online help**
- **Hypertext (HTML)**

This documentation is produced using DECdocument, which produces output from a single set of sources ensuring complete consistency between each format.

Search Lists

OpenVMS supports the concept of a search list with the following behaviour:

- **Read and update access to an object looks through each entry in the search list until the object is found**
- **Creation is perfomed in the first entry of the search list**

Some objects which form search lists include:

- **Nodes, devices, directories and files**
- **Logical name tables**
- **CDD/Repository dictionaries**
- **CMS libraries; CMS class search lists are a SysWorks extension**

Some SysWorks uses of search lists include:

- **Variant application development environments.**
- **Developer specific development environments.**

Logical Name Tables

- OpenVMS uses a search list of logical name tables as follows:

  LNM$PROCESS
  LNM$JOB
  LNM$GROUP
  LNM$SYSTEM

- DECwindows/Motif adds the DECW$LOGICAL_NAMES table after the LNM$SYSTEM table.
- SysWorks enhances this by extending the list to:

  LNM$PROCESS
  LNM$JOB
  LNM$USER
  LNM$CONTEXT
  LNM$GROUP
  LNM$SYSTEM
  DECW$LOGICAL_NAMES

- Furthermore, LNM$CONTEXT can also be a search list of the chained contexts.

Logical Name Tables (cont)

- The logical name of the logical name table search list is LNM$FILE_DEV. This logical name is defined in the LNM$DIRECTORIES search list.
- The names in the LNM$FILE_DEV search list may be logical names which have an equivalence of the actual logical name tables.
- The LNM$USER logical name table is a system wide logical name table which is generally populated the first time that a user logs into a node.
- The LNM$CONTEXT logical name table is switched to the appropriate application environment or group logical name table when the CONTEXT command is issued.
- When a user is in their home environment, no LNM$CONTEXT logical name table is used.

SysWorks Directories

- **SWRK_DAT_DIR**
  Location of data files and generated command procedures.
- **SWRK_DOC_DIR**
  Location of bookreader, help and PostScript and documentation.
- **SWRK_LCL_DIR**
  Location of site specific command procedures.
- **SWRK_SFT_DIR**
  Location of software.

System User Classes

- In system and turnkey installations:
  - + An identifier of the form S_*system-user-class* is used as security
  - + User management places users in one or more system users classes
  - + Privileges and/or quotas may be associated with system users classes
- In public installations, they may be defined in SWRK_LCL_DIR:SWRK_ALTERNATIVE_IDENTIFIERS.DAT
- Each system user class has a directory in SWRK_DAT_ROOT:
- A site specific directory may be created in SWRK_LCL_ROOT:

System User Classes (cont)

The following standard pre-defined System User Classes exist:

System User Class	Usage
ACMS	ACMS user - provides access to ACMS.
ALL	All users - provides base level functions for all users.
ALLIN1	All-In-1 user - provides access to All-In-1.
APPLICATION	Applications - provides base level functions for all applications.
BASE	Automatically granted to provide basic OpenVMS authorization details.
CAPTIVE	Captive users - provides reduced base level functions for captive users.
DBA	Database Administrator - allows a users to use the SysWorks database administration features.
DCL	DCL user - allows a user access to DCL. Note that this system user class is only necessary when the USER system user class does not allow access to DCL by default.
DEVELOPER	Allows a users to use the SysWorks development features.
GROUP	Groups - provides base level functions for all groups.
OPERATOR	Operator - almost full read access and limited read/write access to all information.
PATHWORKS	Pathworks user - provides access to Pathworks including LAN Manager, PCSA and Macintosh variants.
PRINTER_CONTROLLER	Printer controller - may start and stop print queues and print jobs.
SYSTEM_MANAGER	System manager - full read/write access to all information.
USER	User recognized by SysWorks - All users should be a member of this system user class - the exception being Digital and other third party product usernames.

System User Class	Usage
USER_REGISTRAR	User registrar - may add, create, delete and remove users from clusters and the security domain. Note that although a user registrar may register any user, they may only grant access to system user classes which they are a member of. Thus only members of the SYSTEM_MANAGER system user class may register other system managers. The exception to this rule is that a member of the SYSTEM_MANAGER system user class is an implicit member of all system user classes, so they can grant a user access to any system user class.

Part II

User Environment

Logging In

Login command procedures

- SYS$MANAGER:SYLOGIN.COM
- SYS$LOGIN:LOGIN.COM
- SYS$MANAGER:DECW$SYLOGIN.COM
- SYS$LOGIN:DECW$LOGIN.COM

Other details

- SysWorks SYLOGIN.COM should be executed in SYS$MANAGER:SYLOGIN.COM or SYS$LOGIN:LOGIN.COM
- In DECwindows/Motif, the entire session is logged in:

  ```
  SYS$LOGIN:DECW$SM.LOG
  ```

- In PATHWORKS DECwindows (or eXcursion), the entire session is logged in:

  ```
  SYS$LOGIN:NETSERVER.LOG
  ```

SysWorks Login Actions

- **Determine exact process mode and variant.**
- **Define various symbols including:**

  CMD_DEVICE
  CMD_GUESSED
  CMD_MOTIF
  CMD_NODE
  CMD_SM
  CMD_TERMINAL
  CMD_TYPE
  CMD_VT300
  CMD_WINDOW

- **Create a user logical name table if not already present. If possible, this is made a permanent system table.**
- **Populate user logical name table if not populated. The logical name *username*_STATE with a value of AVAIABLE indicates this.**
- **Move to saved or default environment.**

Process Modes

- **Batch**
- **Interactive**
  Terminal Devices include LT, TT, TX and VT
  Window Terminal Devices include FT and TW
  Window Devices include MB and WS)
- **Other (Detached)**
- **Network**
  MX$SERVER and PCX$SERVER similar to an interactive window

CMD_*xxxx* Symbols

- **CMD_DEVICE**
- **CMD_GUESSED**
- **CMD_MOTIF**
- **CMD_NODE**
  When CMD_WINDOW is true, this symbol contains the name of the node to which we are being displayed)
- **CMD_SM**
- **CMD_TERMINAL**
  SET TERMINAL commands should only be executed if this symbol is true
- **CMD_TYPE**
  If CMD_TERMINAL is true, this is set to a value such as VT300
- **CMD_VT300**
- **CMD_WINDOW**

User Directories

All these sub-directories are of the form:

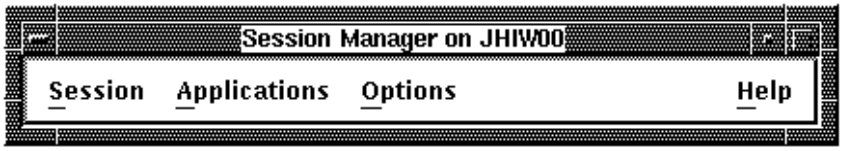
DISK_USER:[*username.sub-dir*]

Table 1: User Directory Usage

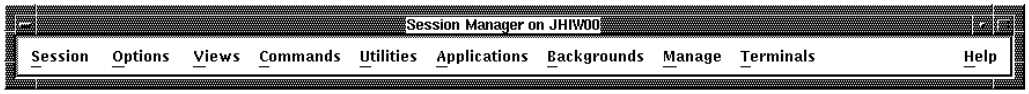
Directory	Code	Usage
DECwindows	DECW	Used instead of SYS$LOGIN: for DECW$USER_DEFAULTS.
	DECW.DENSITY_*dpi*	Used before the [.DECW] sub-directory for DECW$USER_DEFAULTS for a display of the indicated density. Typical *dpi* values include 75 and 100.
	DECW.VAX_*type*	Used before the [.DECW] and [.DECW.DENSITY_*dpi* sub-directories for DECW$USER_DEFAULTS for a VAXstation of the indicated type.
DECwindows Application	DECW.*dwapplcod*	Used before the above directories for DECW$USER_DEFAULTS for the indicated DECwindows/Motif application.
All-In-1	A1	Used as the users All-In-1 directory.
Mail	MAIL	Use as the users OpenVMS Mail and DEC MAILworks directory.
Data	DAT	Used instead of SYS$LOGIN: for user data files and generated DCL command procedures such as SWRK_DECW_LOGICALS_*node*.SBC.
Software	SFT	Used instead of SYS$LOGIN: for user software such as DCL command procedures such as HOME.COM and LOGICALS.COM.

Session Manager

DECwindows/Motif default Session Manager menu bar



SysWorks extended Session Manager menu bar



How This Happens

- **Session Manager**
- **FileView**
- **Resource Files**
  Used by all Xwindow applications.
- **Profile Files**
  Used by the Session Manager / FileView
- **DECW$USER_DEFAULTS**
  Search list of [.DECW] sub-directory, or if not present, SYS$LOGIN: followed by SWRK_VUE_LIBRARY:
- **VUE$LIBRARY**
  Search list including SWRK_VUE_LIBRARY:
- **Window application LOGIN, ENTER and EXIT procedures**

Session Manager and FileView

- In DECwindows/Motif, the Session Manager and FileView are the same shareable image
- New View from Session pulldown menu creates a view in the same process
- FileView from Application pulldown menu creates a view in a new process
- From DCL

Example

```
$ vue := $vue$master
$ vue
```

or

```
$ mcr vue$master
```

Resource Files

- Found in DECW$USER_DEFAULTS or DECW$SYSTEM_DEFAULTS
- File type of .DAT
- Comments either ! or #
- Resource names and values
- Documented in Appendix E of the DECwindows/Motif User's Guide
- Also found in SWRK_VUE_LIBRARY

Example

```
!
!  DECwindows resources:
!       DECW$MAIL
!
!  Purpose:
!       Provide the resources for the DECwindows Mail.
!
Mail.mainPaneWindow.workPaneArea.upperArea.height: 61
Mail.x:    35
Mail.y:    75
Mail.width:   605
Mail.height:   305
Mail.userProfile.iconPostText:
Mail.userProfile.svn:  yes
Mail.userProfile.deliverFolder: New Mail
Mail.userProfile.autoRefile: no
Mail.userProfile.numDrawers: 1
Mail.mailDrawer.1display: MAIL
Mail.mailDrawer.1filespec: *MAIL
Mail.initialState:  3
MailRead.x:   35
MailRead.y:   420
MailRead.width:   605
MailRead.height:  425
MailRead*envelopeText.rows: 5
MailSend.x:   35
MailSend.y:   435
MailSend.width:   605
MailSend.height:  425
```

Profile Files

- Found in VUE$LIBRARY
- File type of .VUE$DAT
- Indexed files managed by Session Manager or File-View
- Use Utilities Any Item... and select Create Public Profile File to edit a profile file.
- Contains information on:
  - Menu items
  - Menus
  - Views
  - File Types
  - Etc...
- Documented in Chapter 14 of the DECwindows/Motif User's Guide
- Also found in SWRK_VUE_LIBRARY

SWRK_VUE_LIBRARY Logical Name

- **System User Classes**
  S_*system-user-class*
- **Application specific**
  A_*appl_envr_class*
- **Determined in file SWRK_ALTERNATIVE_IDENTIFIERS.DAT in SWRK_LCL_DIR:**
  Each entry is a System User Class or Application Environment User Class identifier name with an optional local identifier name.

Example

Example 1: Typical SWRK_ALTERNATIVE_IDENTIFIERS.DAT

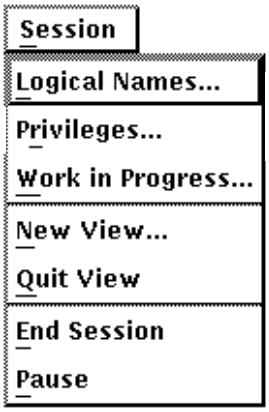
```
!++
!
!  File:
!       SWRK_ALTERNATIVE_IDENTIFIERS
!
!  Purpose:
!       Provide ACME Widgets public
!       System User Classes and Application
!       Environment User Classes
!
!  History:
!       20-Aug-1992 by Simon L. Jackson
!           Initial version
!
!--
S_ALLIN1
S_DEVELOPER
S_USER
A_FIN_DEV_MGR  CAPSUP
A_FIN_MNT_MGR  CAPSUP
A_MAN_DEV_MGR  DSSSUP
A_MAN_MNT_MGR  DSSSUP
A_RIM_DEV_MGR  FRSSUP
A_RIM_MNT_MGR  FRSSUP
```

Example

Example 2: Typical SWRK_VUE_LIBRARY logical name

```
$ show logical SWRK_VUE_library
   "SWRK_VUE_LIBRARY" = "SWRK_DAT_ROOT:[ALL]" (LNM_JACKS
        = "SWRK_DAT_ROOT:[DEVELOPER]"
        = "SWRK_DAT_ROOT:[OPERATOR]"
        = "SWRK_DAT_ROOT:[PATHWORKS]"
        = "SWRK_DAT_ROOT:[SYSTEM_MANAGER]"
        = "SWRK_DAT_ROOT:[USER]"
        = "DISK_DEV3:[SWRK.DAT]"
        = "DISK_PROD2:[SWRK.DAT]"
        = "DISK_APPL3:[SWPUB.DAT]"
        = "DISK_DEV3:[SWPUB.DAT]"
        = "DISK_DTST3:[SWPUB.DAT]"
        = "DISK_MNT3:[SWPUB.DAT]"
        = "DISK_MTST3:[SWPUB.DAT]"
   "SWRK_VUE_LIBRARY" = "SWRK_DAT_ROOT:[NEVER]" (LNM$SYS
```

Session
Pulldown Menu



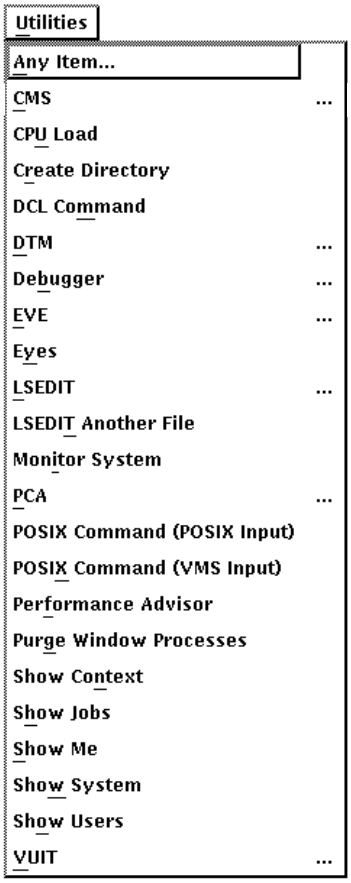
Options
Pulldown Menu

Views

- **Many attributes may be saved in a view.**
- **Select Save View form the Options pull down menu.**
- **Standard Views include:**
  - **Session Manager**
  - **Session Manager as FileView**
  - **Startup**
- **Links to utilities and applications such as LSEDIT via device and directory**

Commands
Pulldown Menu

Utilities
Pulldown Menu



Network Commands

- **DECTERM icon-name window-name**
  Optimized for local node
- **NETTERM node[,...] icon-name window-name**
- **PWD node[,...]**
  No alias or network access logical used
- **RW node,[...] command**
  Command in network process context the DECnet tast-to-task link is left open.
- **RWI node[,...] command**
  Command in interactive process context no DECnet task-to-task link left open.
- **TELL node[,...] command**

Network Commands (Cont)

- **To access a cluster or node, the user must have either a proxy at the destination or a logical of the form:**

  ```
  $ define user_node node"user password"::
  ```

  where user is the users username, node is the destination node, and password is the password at the destination node.

- **Logicals of this form may be defined in:**

  ```
  SYS$LOGIN:SWRK_DEFINE_NETWORK_LOGICALS.COM
  ```

- **Output logged in**

  ```
  SYS$LOGIN:NETSERVER.LOG
  ```

  at the remote node.

Node Specifiers

All the *node[,...]* parameters may be substituted with one of the following keywords:

- **ALL**

  The following logical name specifies the meaning of the ALL keyword:

  ```
  SWRK_NET_CMD_ALL_TYPE
  ```

- **CLUSTER**

  All nodes within the current cluster as defines by the logical name:

  ```
  SYS$CLUSTER_NODES
  ```

- **CLUSTERS**

  All clusters within the users network as defined by the logical name:

  ```
  SYS$CLUSTERS
  ```

- **NODES**

  All nodes within the users network as defined by the logical name:

  ```
  SYS$NODES
  ```

Part III

Application Environment

Environments

- USER
- GROUP
- APPLICATION
  - + Common (APPL)
  - + Development (DEV and FDEV)
  - + Development Testing (DTST)
  - + Maintenance (MNT)
  - + Maintenance Testing (MTST)
  - + Production (PROD and PRD)
  - + Training (TRNG)

Logical Disks

Each environment has an associated logical disk structure, the logical name of which has the form:

DISK_*envr*:

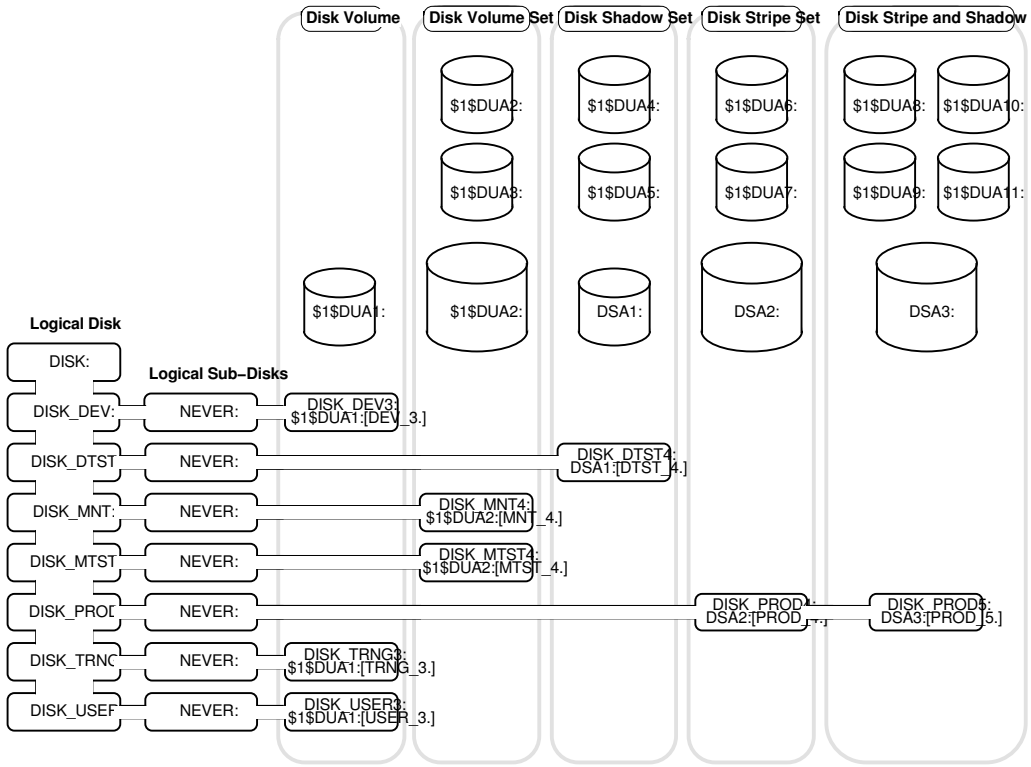
This logical name is a search list of logical sub-disks, each of which is a root directory on a disk volume.

Example

```
$ show logical disk_dev
   "DISK_DEV" = "NEVER:" (LNM$SYSTEM_TABLE)
        = "DISK_DEV3:"
        = "DISK_DEV4:"
1  "DISK_DEV3" = "$1$DUA2:[DEV_3.]" (LNM$SYSTEM_TABLE)
1  "DISK_DEV4" = "$1$DUA3:[DEV_4.]" (LNM$SYSTEM_TABLE)
```

- **The** NEVER: **placeholder prevents creation of files directly in** DISK_DEV:
- **In public installations, this structure must still exist, usually by setting alias directory entries and defining the logical disk structure in the site specific login procedure.**
- **LOGICALS.COM procedures for defining logical names at boot or login time parse directories and files to eliminate the search list.**

Overview of Disk Structure



Changing Contexts

The CONTEXT command includes the following sub-commands:

- **APPLICATION *appl envr {vrnt|vrsn}***

  Change the context to the indicated application environment. May use ENTER.COM, EXIT.COM, LOGICALS.COM.

- **ENVIRONMENT *envr {vrnt|vrsn}***

  Change the context to the indicated environment of the current application. May use ENTER.COM, EXIT.COM, LOGICALS.COM.

- **GROUP *group***

  Change the context to the indicated group. May use ENTER.COM, EXIT.COM, LOGICALS.COM.

- **HOME**

  Change the context to the users HOME area. May use HOME.COM.

- **VARIANT *vrnt***

  Change the context to the indicated variant of the current application environment. May use ENTER.COM, EXIT.COM, LOGICALS.COM.

- **VERSION *vrsn***

Change the context to the indicated version of the current application environment. May use ENTER.COM, EXIT.COM, LOGICALS.COM.

Each of the DCL command procedures which may be used during the move resides in the associated software directory.

Context Scopes

The application environments also have a scope attribute.

The supported scopes include:

- **COMMON**
- **SPECIFIC**
- ***other*)**

These are supported by the following qualifiers on various commands:

/COMMON
/SPECIFIC
/SCOPE=*other*)

Directories with an extra search list item for the non COMMON scopes include:

Architecture Independent Library
Data
Documemtation
Library
Runtime
Software
Testing
Work

Context Command Procedures

- **LOGICALS.COM**

  Used to define application environment specific logical names. See SWRK_SFT_DIR:LOGICALS_APPL_ENVR.TEMPLATE for a complete example.

- **ENTER.COM**

  Used to define application specific symbols and job and/or process logical names and inter-application dependencies.

- **EXIT.COM**

  Used to delete application environment specific items defined in ENTER.COM. No need to remove application inter-dependencies.

Application Directories

Application directories are of the form:

DISK_*envr*:[*appl*.*dir-cod*]

Table 2: Application Directory Usage

Directory	SysWorks Default	APPL	FDEV	DEV	MNT	DTST	MTST	PROD TRNG
Software	SFT	No	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.SFT	No	No	Yes	No	Yes	No	No
	vrsn.SFT	No	No	No	Yes	No	Yes	No
Developer specific software	SFT.*user*	No	Yes	Yes	Yes	No	No	No
	VAR*vrnt*.SFT.*user*	No	No	Yes	No	No	No	No
	vrsn.SFT.*user*	No	No	No	Yes	No	No	No
Documentation	DOC	No	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.DOC	No	No	Yes	No	Yes	No	No
	vrsn.DOC	No	No	No	Yes	No	Yes	No
Developer specific documentation	DOC.*user*	No	Yes	Yes	Yes	No	No	No
	VAR*vrnt*.DOC.*user*	No	No	Yes	No	No	No	No
	vrsn.DOC.*user*	No	No	No	Yes	No	No	No
Kit	KIT	Yes	No	No	No	Yes	Yes	No
Distribution	KIT.*vrsn*	Yes	No	No	No	No	No	No
Data	DAT	Opt	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.DAT	No	No	Yes	No	Yes	No	No
	vrsn.DAT	No	No	No	Yes	No	Yes	No
Journal	JNL	No	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.JNL	No	No	Yes	No	Yes	No	No
	vrsn.JNL	No	No	No	Yes	No	Yes	No
Runtime	RUN	No	Yes	Yes	Yes	Yes	Yes	Yes

Table 2 (Cont.): Application Directory Usage

Directory	SysWorks Default	APPL	FDEV	DEV	MNT	DTST	MTST	PROD TRNG
	VAR*vrnt*.RUN	No	No	Yes	No	Yes	No	No
	vrsn.RUN	No	No	No	Yes	No	Yes	No
User runtime	RUN.*user*	No	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.RUN.*user*	No	No	Yes	No	Yes	No	No
	vrsn.RUN.*user*	No	No	No	Yes	No	Yes	No
Repository	CDD	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.CDD	No	No	Yes	No	Yes	No	No
	vrsn.CDD	No	No	No	Yes	No	Yes	No
Architecture independent library	AIL	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.AIL	No	No	Yes	No	Yes	No	No
	vrsn.AIL	No	No	No	Yes	No	Yes	No
Developer specific architecture independent library	AIL.*user*	No	Yes	Yes	Yes	No	No	No
	VAR*vrnt*.AIL.*user*	No	No	Yes	No	No	No	No
	vrsn.AIL.*user*	No	No	No	Yes	No	No	No
Library	LIB	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.LIB	No	No	Yes	No	Yes	No	No
	vrsn.LIB	No	No	No	Yes	No	Yes	No
Developer specific library	LIB.*user*	No	Yes	Yes	Yes	No	No	No
	VAR*vrnt*.LIB.*user*	No	No	Yes	No	No	No	No
	vrsn.LIB.*user*	No	No	No	Yes	No	No	No
Compatibility Dictionary	AIL.DMU	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.AIL.DMU	No	No	Yes	No	Yes	No	No
	vrsn.AIL.DMU	No	No	No	Yes	No	Yes	No
SCA Library	LIB.SCALIB	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.LIB.SCALIB	No	No	Yes	No	Yes	No	No
	vrsn.LIB.SCALIB	No	No	No	Yes	No	Yes	No
Work	WRK	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.WRK	No	No	Yes	No	Yes	No	No

Table 2 (Cont.): Application Directory Usage

Directory	SysWorks Default	APPL	FDEV	DEV	MNT	DTST	MTST	PROD TRNG
	vrsn.WRK	No	No	No	Yes	No	Yes	No
User work	WRK.*user*	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.WRK.*user*	No	No	Yes	No	Yes	No	No
	vrsn.WRK.*user*	No	No	No	Yes	No	Yes	No
Scratch	SCR	No	Yes	Yes	Yes	Yes	Yes	Yes
	VAR*vrnt*.SCR	No	No	Yes	No	Yes	No	No
	vrsn.SCR	No	No	No	Yes	No	Yes	No
Source	SRC	Yes	Yes	Opt	No	No	No	No
CMS library	SRC.CMSLIB	Yes	Yes	Opt	Yes	No	No	No
Test	TST	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.TST	No	No	Yes	No	Yes	No	No
	vrsn.TST	No	No	No	Yes	No	Yes	No
DTM library	TST.DTMLIB	No	Yes	Yes	Yes	Yes	Yes	No
	VAR*vrnt*.TST.DTMLIB	No	No	Yes	No	Yes	No	No
	vrsn.TST.DTMLIB	No	No	No	Yes	No	Yes	No

Changing Directory Code Defaults

- **SDC_*xxx***

  For directory code *xxx* the symbol SDC_*xxx* may contain the site specific alternative directory code.

  For example if the symbol SDC_SFT has a value of EXE, software will be located in [.EXE] directories.

  This symbol is normally defined in the site specific pre login command procedure.

- **DIR_*xxx***

  For directory code *xxx* the symbol DIR_*xxx* may contain the site or application specific alternative directory code.

  For example if the symbol DIR_SFT has a value of SN_CAP_EXE:, software will be located in that directory. This symbol overrides any SDC_*xxx* symbol.

  This symbol is normally defined in the application environment ENTER.COM command procedure and deleted by EXIT.COM.

Editing

Table 3: Editors and Modes

Editor	Direct	Subprocess	Window
EDT	Yes	No	No
EVE	Yes	Yes	Yes
LSEDIT	Yes	Yes	Yes, with DCL connection

Editing

Table 4: Editor Key Extensions

Key	Action
F10	Exit - behavior depends upon mode.
F17	Return to DCL - only used with the subprocess mode
Gold-=	Toggle between one window and split windows
Gold-+	Split the current window.
Gold–	Remove the current window
Gold-[	Move window one screen to the left.
Gold-{	Move window to left of the buffer.
Gold-’	Move window to cursor.
Gold-}	Move window to right of the buffer.
Gold-]	Move window one screen to the right.
Gold-B	Toggle between the current buffer and the list of buffers.
Gold-C	Close the current buffer.
Gold-D	Delete the current buffer.
Gold-E	Same as F10.
Gold-F	Fill the currently selected region or paragraph.
Gold-M	Toggle between the current buffer and the message buffer.
Gold-Q	Quit - behavior depends upon mode.
Gold-S	Toggle between the current buffer and the list of system buffers.
Gold-T	Toggle between narrow and wide windows.
Gold-W	Write out the current buffer.

Editing

Table 5: Editor Key Mode Differences

Mode	Key	Semantics
Direct	F10, Gold-E	Close all buffers and exit back to DCL.
	Gold-Q	Delete all buffers and exit back to DCL.
Subprocess	F10, Gold-E	Close current buffer and return to DCL.
	F17	Return to DCL.
	Gold-Q	Delete current buffer and return to DCL.
Window		Selection causes implied APPLICATION command.
	F10, Gold-E	Write current buffer.
	Gold-Q	Delete current buffer.

LSE Extensions

- When the EDIT command is used from DCL, the most recently used DECwindows/Motif LSE session will be automatically restored and reactivated. If none is available, a sub-process LSE session will be created.
- Various LSE commands have their function changed in line with SysWorks extensions. These include:
  - + CREATE ELEMENT
  - + FETCH
  - + REPLACE
  - + RESERVE
  - + UNRESERVE
- The following SysWorks extension is not yat availble directly from LSE:
  - + PROMOTE GENERATION

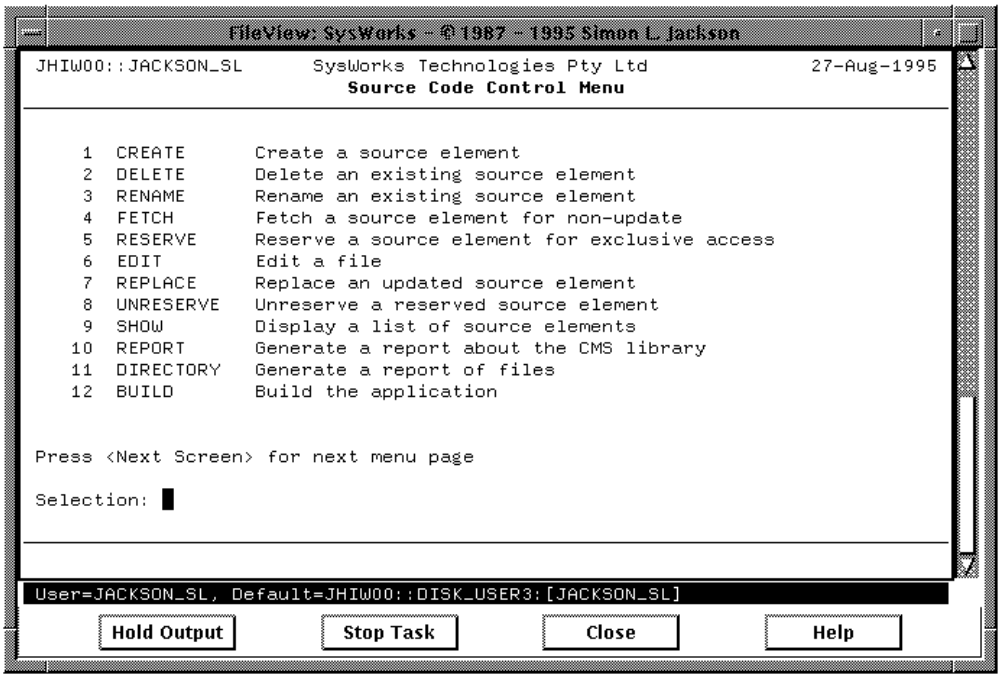
Part IV

Application Development

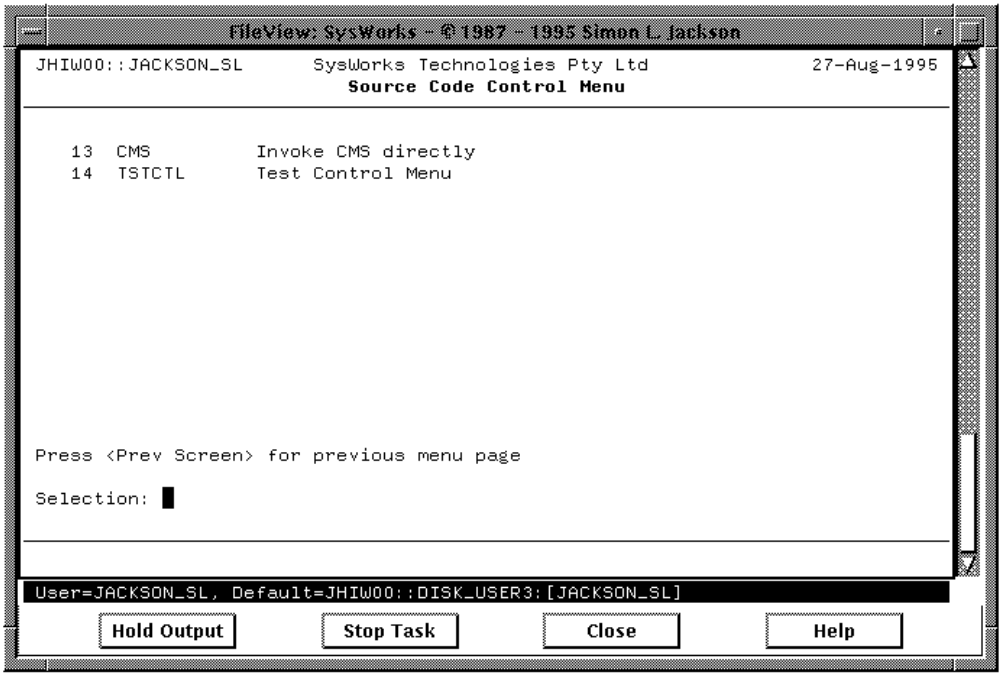
CMS Concepts

- **ELEMENT**
  A source file
- **GROUP**
  A collection of zero or more elements
- **GENERATION**
  A version of a source file
- **CLASS**
  A collection of zero or more generations
- **PATH (SysWorks extension)**
  A search list of generation expressions

Source Control



Source Control (Cont)



Source Control (Cont)

- Acts as a front end for CMS.
- The keyword ELEMENT in commands such as CREATE does not need to be supplied; the second parameter defaults to ELEMENT when it is not CLASS, ELEMENT or GROUP.
- If an *appl*_CMS_PATH logical name is defined (typically in the application logical name table), the behaviour of CMS commands change to include handling the classes and/or generations specified.
- If the *appl*_CMS_PATH logical name is present, an *appl*_CMS_VARIANT logical name may further change the behaviour of the CMS commands.
- The BUILD command is the same as the DCL BUILD command, except that a series of questions are asked in order to build up the command's qualifiers.

SRCCTL vs CMS

Behaviour changes with a *appl*_CMS_PATH

- **CREATE ELEMENT**

  Creates the element. If a variant was specified, it is reserved and replaced with the variant. The appropriate generation is then inserted into the first class in the path.

- **DELETE ELEMENT**

  Removes a generation from the first class in the path in which one is found. It does not actually delete the element.

- **RENAME ELEMENT**

  Removes a generation from the first class in the path in which one is found. If the new name already exists, creates a new or variant generation of it is created and inserted into the first class in the path. Otherwise, a new element is created, and the new generation is inserted into the first class in the path as per the CREATE command.

- **FETCH**

  Fetches the first generation from found in the path.

SRCCTL vs CMS (cont)

- **RESERVE**

  Reserves the first generation from found in the path. This must be the latest in its branch otherwise a promotion required error will be produced.

- **EDIT**

  Edits a reserved source. Primarily useful from within a character cell terminal environment. Also accepts a wildcard file specification, and 'feeds' through each of the files to be edited.

- **REPLACE**

  Replaces a reserved generation. There is no effective change of operation from CMS.

- **UNRESERVE**

  Unrserves a reserved generation. There is no effective change of operation from CMS.

- **SHOW ELEMENT**

  Shows the generations within the path rather than all elements.

Development Commands

- **BUILD[/qualifiers] [target]**
- **CHANGE[/qualifiers] source-file[,...] search-string[,...] replace-string[,...]**
- **COMPARE[/qualifiers] new-area[/qualifiers] old-area[/qualifiers]**
- **COMP*ILE[/qualifiers] source-file[,...]**
- **CVTMMS source-file[,...] [target-mms-script]**
- **DO[/qualifiers] command**

Commands with Context

Commands which support context include:

- **BUILD**
- **COMPILE**
- **DO**

Qualifiers include:

```
/AFTER
/BATCH
/DETACHED
/INTERACTIVE
/KEEP
/LOG
/SUBPROCESS
/SYNCHRONIZE
```

Commands with Memory

Commands which support memory include:

- **BUILD**
- **COMPILE**

Qualifiers include:

```
/CLUSTER
/DEFAULTS
/MEMORY
/NODE
/SAVE
/UNSAVE
```

BUILD

Format:

```
BUILD[/qualifiers] [target]
```

Multiple phases (with the /PHASES qualifier) including:

- **UPDATE**

  Copy any sources in the user work directory which are newer than the last BUILD.LOG into the work directory.

- **FETCH**

  Bring the work directory into line with the CMS class by fetching any newer sources from the CMS library

- **SETUP**

  Perform any special operations needed to perform the SCA and/or RULES phases such as building MMS generators.

- **SCAN**

  Scan the CMS library and produce a set of dependecy files needed during the RULES and DESCRIP phases. Several abstract targets are generated such as SCRIPT_LIST and FORM_LIST.

  Resulting MMS scripts are in the library directory.

BUILD (cont)

- **RULES**

  Convert most sources into MMS scripts. These are appended together to produce *appl*_APPENDS which is used during the DESCRIP phase.

- **DESCRIP**

  Actual compile, link and build actions. This is the traditional MMS or makefile.

- **KITBLD**

  Takes the resultant software and creates a kit from it.

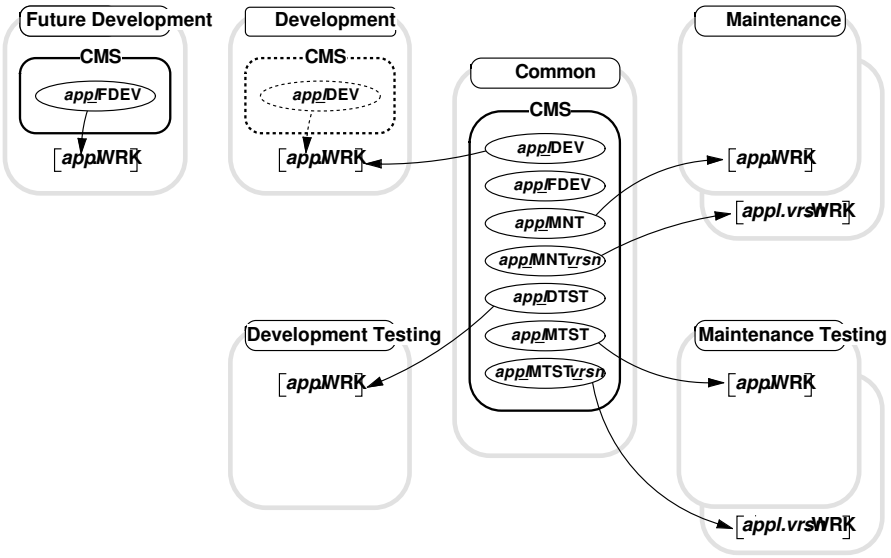
BUILD (cont)

Qualifiers include:

```
/CMS
/DEBUG
/DESCRIP
/FETCH
/FROM_SOURCES
/KITBLD
/LIST
/MAP
/MMS
/PHASES
/RULES
/UPDATE
```

BUILD (cont)

Figure 1: BUILD FETCH Phase Flow



BUILD SCAN Phase Results

Table 6: SCAN Phase Generated Files

Generated File	Usage
*appl*_CONSTRAINTS.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_table*_CK*n*.SQL, *appl_table*_FK*n*.SQL and *appl_table*_PK.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_DOCUMENTATION.MMS	A dependency list of all the documentation targets.
*appl*_DOMAINS.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_domain*_DOM.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_DOMAIN_SDML_ LIST.MMS	A dependency list of all the generated SDML domain documentation sources.
*appl*_FORM_LIST.MMS	A dependency list of all the generated DECforms form MMS scripts, so that a generated linker options file may be regenerated when any DECforms form is modified. This regeneration is typically performed near the end of the RULES phase.
*appl*_HELP.MMS	A dependency list of all the help targets.
*appl*_INDICES.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_index*_IDX.SQL or *appl_index*_PIDX.SQL or *appl_index*_SIDX*n*.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_LINK_OPT_LIST.MMS	A dependency list of all the Linker options sources.
*appl*_MSGHLP.MMS	A dependency list of all the message help targets.
*appl*_PROTECTIONS.SQL	An SQL source which will apply a standard protection to each appropriate object in the database.

Table 6 (Cont.): SCAN Phase Generated Files

Generated File	Usage
*appl*_ROUTINES.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_routine*_RTN.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_SCHEMA.MMS	A dependency list of all the Rdb database schema sources. This script is usually included in the application DESCRIP.MMS script.
*appl*_STORAGE_ MAPS.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_storage_map*_ SM.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_TABLES.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_table*_TBL.SQL. Note that SQL constraints definitions are embedded within the table definition sources.
*appl*_TABLE_SDML_ LIST.MMS	A dependency list of all the generated SDML table documentation sources.
*appl*_TARGETS.MMS	A dependency list of all the targets.
*appl*_TASK_LIST.MMS	A dependency list of all the generated ACMS task MMS scripts, so that a generated ACMS task group may be regenerated when any ACMS task is modified. This regeneration is typically performed near the end of the RULES phase.
*appl*_TRIGGERS.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_trigger*_TRG.SQL or *appl_trigger*_TRGR.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_VIEWS.SQL	An SQL script containing a list of all the sources with a file name and type of the form *appl_view*_VIEW.SQL or *appl_view*_VW.SQL. The @ character precedes each source file specification so that this procedure can be used from within the SQL utility.
*appl*_VIEW_SDML_ LIST.MMS	A dependency list of all the generated SDML view documentation sources.

BUILD RULES Phase Results

Table 7: RULES Phase Generated Files

Generated File	Usage
*appl*_DEPENDENCIES.MMS	An MMS script containing all the generated dependencies used to build the application.

CHANGE

Format:

```
CHANGE[/qualifiers] -
 source-file[,...] -
 search-string[,...] -
 replace-string,[...]
```

- **Simple search and replace on source files.**
- **Each occurence of a search string is replaced by its corresponding replace string.**
- **If there are insufficient replace strings, the last one is used multple times. For example:**

  ```
  $ change a.txt a1,a2,a3 b1,b2
  ```

  would result is all occurences of `a1` **being changed to** `b1` **and all occurences of** `a2` **and** `a3` **being changed to** `b2`**.**

COMPARE

Format:

```
COMPARE[/qualifiers] -
 new-area[/qualifiers] -
 old-area[/qualifiers]
```

- **Compare the revision dates of files or CMS generations.**
- **Generates a DCL command procedure to bring them in line.**
- **Used in the change control TRIAL ask.**
- **Each of the two parameters may be one of:**
  - **file-dir**
  - **cms-dir/CMS**
  - **cms-dir/CMS/PATH=cms-class-list**
- **Uses a rule based system to determine the actions to perform.**
- **Over 150 rules used.**
- **Other qualifiers include:**

  ```
  /BOTH
  /OUTPUT
  ```

- **Useful to check for accidental changes or additions to work directories etc.**

COMPILE

Format:

```
COMPILE[/qualifiers] source-file[,...]
```

- **Compiles a source to produce its associated target.**
- **Other qualifiers include:**

```
/AFTER
/BATCH
/DEBUG
/DETACHED
/SUBPROCESS
```

Language Summary

Table 8: Language Support Summary

Language or Product	Source File Types	Compile	Target Directory	Target File Types
ACA Services	.COL, .CRL	Yes	Library	.C, .MMS_INC
			Software	.CO, .CR
	.CRL_IND	No	Library	.CRL, .MMS_INC
ACMS	.ADF, .GDF, .MDF, .TDF	Yes	Architecture Indepen-dent Library	.MMS_INC, .OPT, .OPT_ INC, .TAG_CDD
			Software	.ADB, .MDB, .TDB
	.ADF_INC, .GDF_INC, .MDF_INC, .TDF_INC	No	Architecture Indepen-dent Library	.MMS_INC
Ada[1]	.ADA	Yes	Library	.MMS_INC, .OBJ, .OLB. .TAG_EP
Basic[1]	.BAS	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
C	.C	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
	.H	No	Library	.MMS_INC, .TAG_INC_1, .TAG_INC_2
C++	.CXX	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
	.HXX	No	Library	.MMS_INC, .TAG_INC_1, .TAG_INC_2
	.IXX	No	Library	.MMS_INC, .TAG_INC_1, .TAG_INC_2
CDD	.CDDL, .DDL	Yes	Architecture Indepen-dent Library	.MMS_INC, .TAG_CDD

[1]The MMS generators for this language and its associated associated SQL precompiler generators will be released in a future version of SysWorks Developer.

Table 8 (Cont.): Language Support Summary

Language or Product	Source File Types	Compile	Target Directory	Target File Types
CDD/Plus & CDD/Repository	.CDO	Yes	Architecture Indepen-dent Library	.MMS_INC, .TAG_CDD
Cobol	.COB, .TXT	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP, .TLB
Command Language Definition	.CLD	Yes	Library	.MMS_INC, .OBJ, .TAG_EP
	.CLD_INC	No	Library	.MMS_INC, .TAG_EP
	.CLD_INC_ SRC	Yes	Software	.CLD_INC
Datatrieve	.DTR	Yes	Architecture Indepen-dent Library	.MMS_INC, .TAG_CDD
DECdocument	.GRA, .SDML	Yes	Library	.HLP, .MMS_INC, .TAG_ SDML, .TXT
	.SDML_INC	No	Library	.MMS_INC, .TAG_SDML
			Documentation	.DECW$BOOK, .DECW$BOOKSH .GIF, .HLB, .HTML, .PS, .RELEASE_NOTES
DECwindows	.UIL	Yes	Architecture Indepen-dent Library	.MMS_INC
			Software	.UID
FMS	.FRM	Yes	Library	.MMS_INC, .OBJ
Fortran	.FOR	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
HTTP	.SHTML	Yes	Documentation	.MMS_INC
			Documentation	.HTML

Table 8 (Cont.): Language Support Summary

Language or Product	Source File Types	Compile	Target Directory	Target File Types
	.GIF_CPY, .HTIMAGE_ CPY, .HTM_ CPY, .HTML_ CPY, .HTMLS_ CPY, .HTMLX_ CPY, .IM- AGEMAP_ CPY, .SHTML_ CPY	Yes	Documentation	.GIF, .HTIMAGE, .HTM, .HTML, .HTMLS, .HTMLX, .IMAGEMAP, .SHTML
Java	.JAVA	Yes	Architecture Indepen- dent Library	.MMS_INC
			Software	.CLASS
Linker	.OPT, .OPT_ APP, .OPT_ INC	Yes	Library	.MMS_INC, .OPT_EXP
			Software	.EXE
Macro-32	.MAR, .MLB	Yes	Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
			Software	.MLB
ObjectBroker	.IDL	Yes	Library	.C, .MMS_INC
			Software	.IR
Pascal	.PAS	Yes	Library	.MMS_INC, .OBJ, .OLB, .PEN, .TAG_EP
PL/1[1]	.PLI	Yes	Library	.MMS_INC, .OBJ, .OLB
PowerHouse	.QKS, .QTS, .QZS	Yes	Architecture Indepen- dent Library	.MMS_INC, .TAG_INC
	.QKS_INC, .QTS_INC, .QZS_INC	No	Architecture Indepen- dent Library	.MMS_INC, .TAG_INC
			Software	.QKC, .QTC, .QZC

[1]The MMS generators for this language and its associated associated SQL precompiler generators will be released in a future version of SysWorks Developer.

Table 8 (Cont.): Language Support Summary

Language or Product	Source File Types	Compile	Target Directory	Target File Types
Rdb	.RDO	No	Architecture Independent Library	.CDO_GEN, .MMS_INC, .TAG
	.RBA, .RC, .RCO, .RFO, .RPA	Yes	Architecture Independent Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
	.SAD, .SBA, .SC, .SCO, .SFO, .SPA, .SPL	Yes	Architecture Independent Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
	.SQL	No	Architecture Independent Library	.CDO_GEN, .MMS_INC, .TAG
	.SQLMOD	Yes	Architecture Independent Library	.MMS_INC, .OBJ, .OLB, .TAG_EP
Runoff	.RND	Yes	Architecture Independent Library	.MMS_INC
			Documentation	.DOC
	.RNH	Yes	Architecture Independent Library	.HLP, .MMS_INC
			Documentation	.HLB
	.RNM, .RNR	Yes	Architecture Independent Library	.MMS_INC
			Documentation	.MAN, .RELEASE_NOTES
TDMS	.FRM_TDMS, .RDF	Yes	Architecture Independent Library	.MMS_INC, .TAG_CDD
	.LDF	Yes	Architecture Independent Library	.MMS_INC, .TAG_CDD
			Software	.RLB

Table 8 (Cont.): Language Support Summary

Language or Product	Source File Types	Compile	Target Directory	Target File Types
DECdocument	.GRA, .SDML	Yes	Architecture Independent Library	.BRF, .MMS_INC
			Documentation	.DECW$BOOK, .PS, .RELEASE_NOTES, .TXT
	.SDML_INC	No	Architecture Independent Library	.MMS_INC

CVTMMS

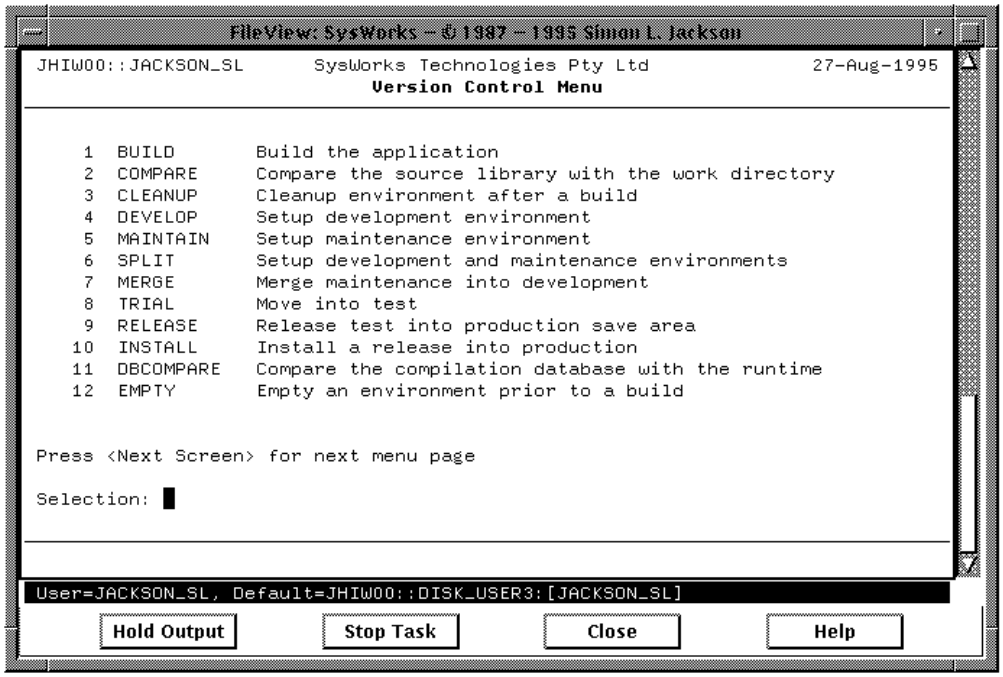
Format:

```
CVTMMS source-file[,...] [target-mms-script]
```

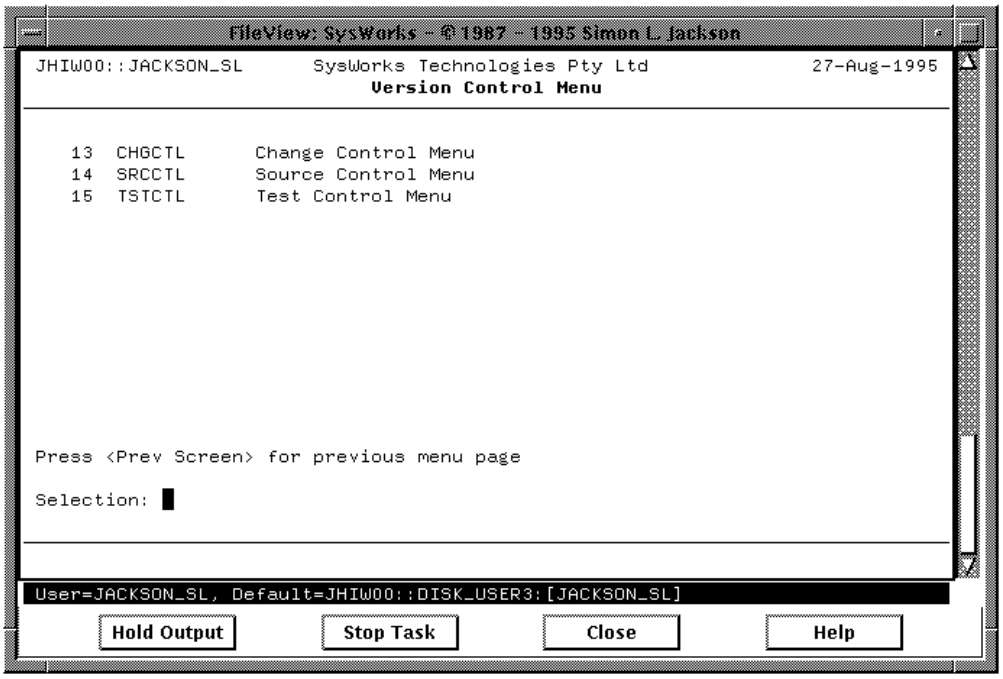
Converts the specified sources into MMS scripts.

The default file type is .MMS_INC.

Version Control

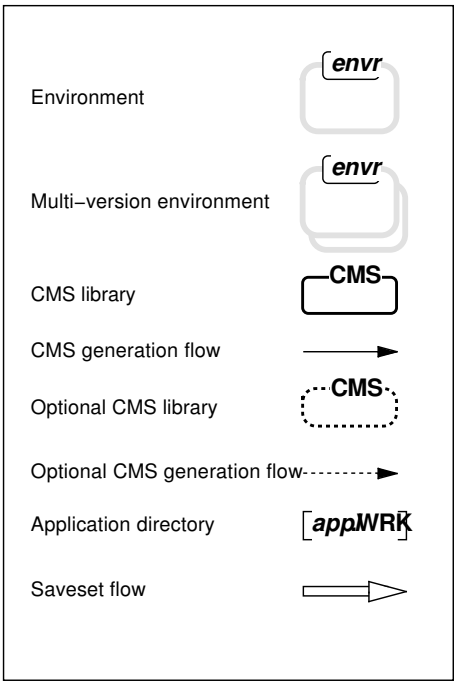


Version Control (cont)



Version Control (cont)

Figure 2: VSNCTL Diagrams Legend



Version Control (cont)

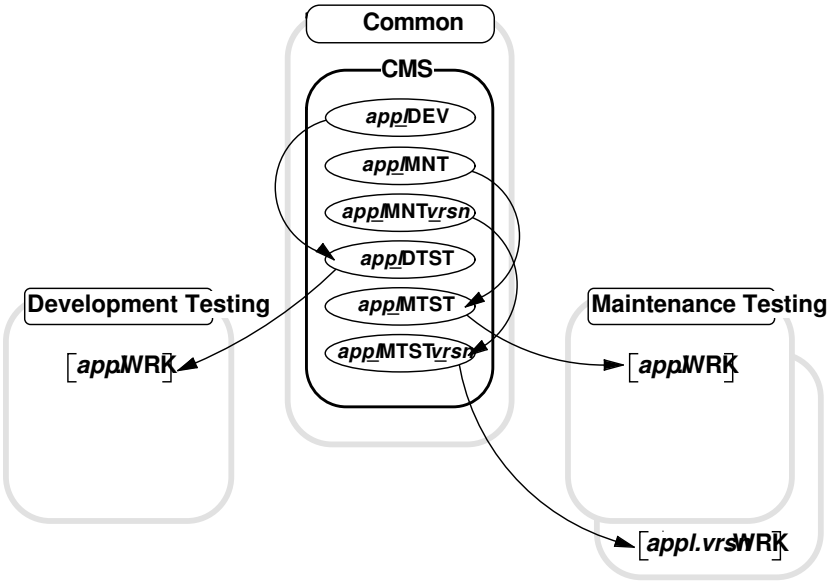
TRIAL

- **Updates target environment CMS library and/or class (uses COMPARE command).**
- **Builds in target environment.**

```
$ vsnctl trial
Target environment [DTST]:
Override CMS reservations (Yes/No) [No]:
Build (Yes/No) [Yes]:
From sources or work area (Source/Work) [Work]:
Execution (Batch/Detached/Online/Subprocess) [Batch]:
Batch queue [SYS$BATCH]:
Execute after [NOW]:
```

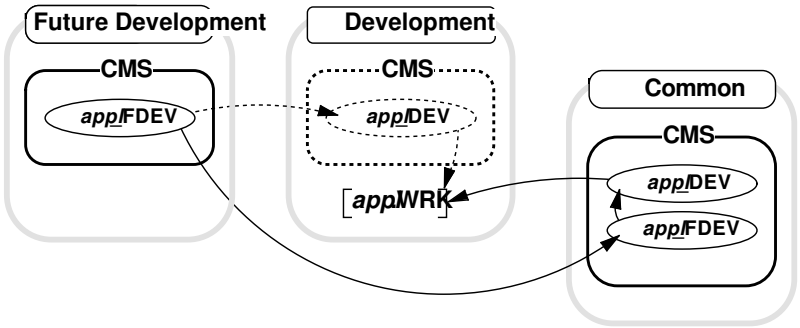
Version Control (cont)

Figure 3: TRIAL flow



Version Control (cont)

Figure 4: TRIAL flow from FDEV to DEV



Version Control (Cont)

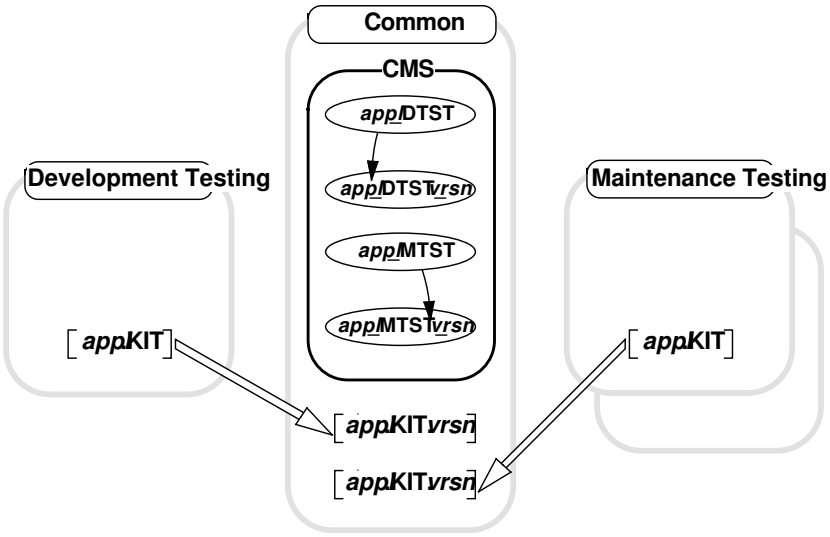
RELEASE

- **Releases a build testing environment source as a production class.**

```
$ vsnctl release
Test environment [DTST]:
Version [V1.0]:
Execution (Batch/Detached/Online/Subprocess) [Batch]:
Batch queue [SYS$BATCH]:
Execute after [NOW]:
```

Version Control (cont)

Figure 5: Release Flow



Version Control (Cont)

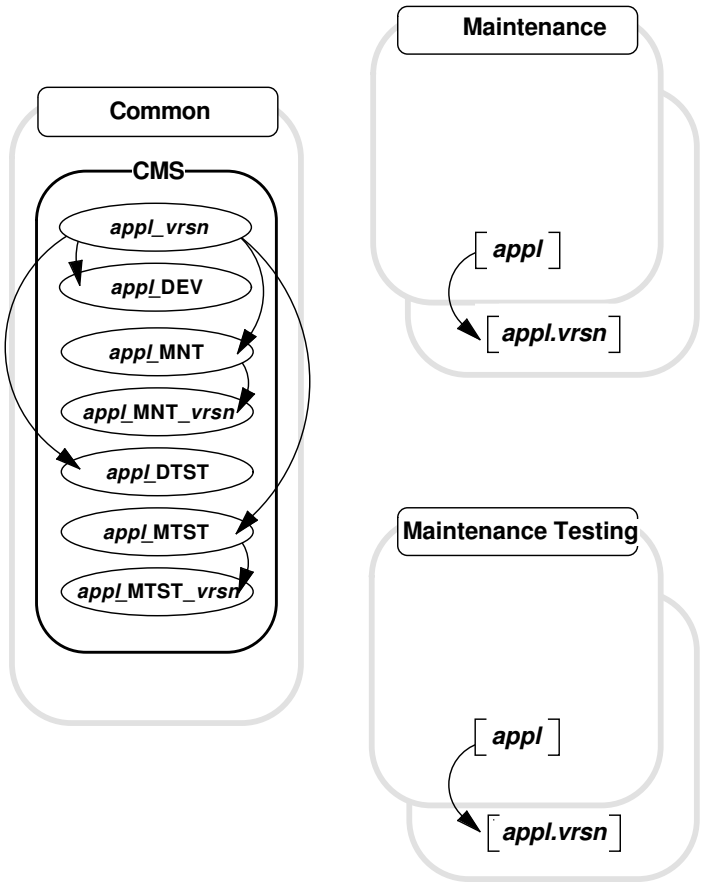
SPLIT

- **Takes a released version and splits it into a development and maintenance version.**
- **Builds the new development and maintenance areas.**

```
$ vsnctl split
Version: V1.0
Development environment [DEV]:
Build (Yes/No) [Yes]:
Development testing environment [DTST]:
Maintenance environment [MNT]:
Build (Yes/No) [Yes]:
Maintenance testing environment [MTST]:
Keep existing maintenance version (Yes/No) [Yes]:
Execution (Batch/Detached/Online/Subprocess) [Batch]:
Batch queue [SYS$BATCH]:
Execute after [NOW]:
```

Version Control (cont)

Figure 6: Split Flow



Version Control (Cont)

MERGE

- **Updates target environment CMS library and/or class (uses COMPARE command).**
- **Leaves unresolved merges reserved for manual resolution.**
- **Doesn't delete sources from the target environment.**

Example:

```
$ vsnctl merge
Target environment [MNT]:
Override CMS reservation check (Yes/No) [No]:
Since [BEGINNING]:
Execution (Batch/Detached/Online/Subprocess) [Batch]:
Batch queue [SYS$BATCH]:
Execute after [NOW]:
```

Version Control (Cont)

INSTALL

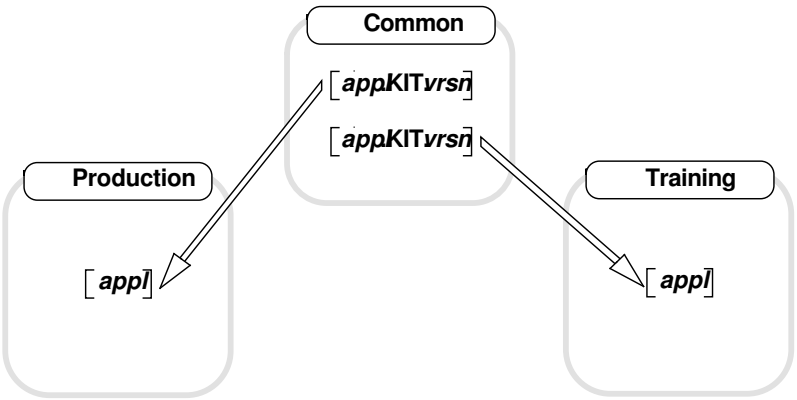
- **Installs releaseed savesets into production or training.**

Example:

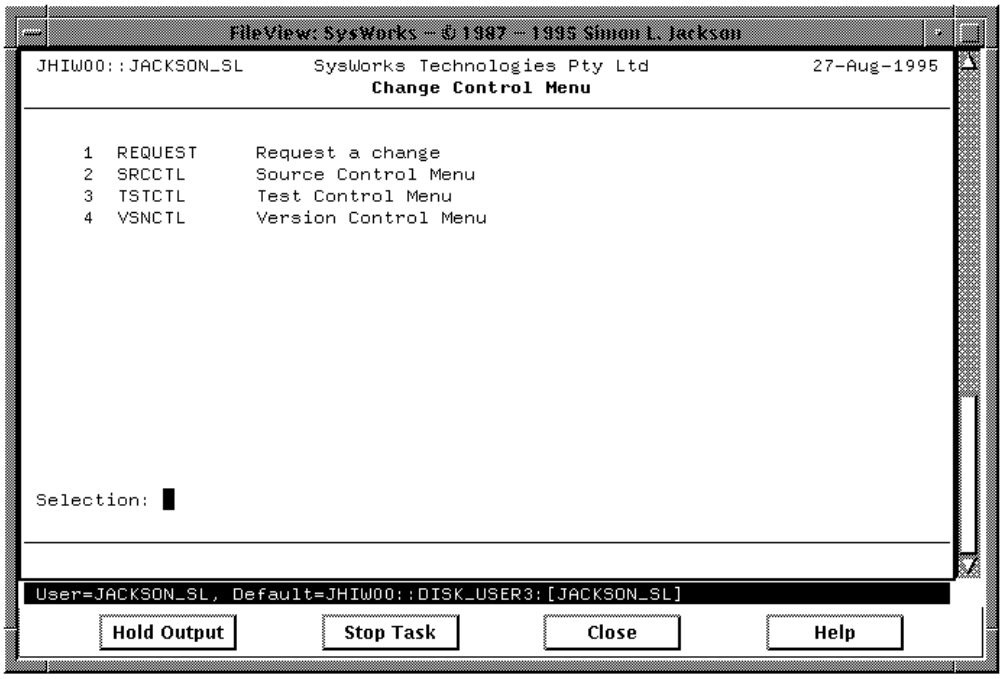
```
$ vsnctl install
Target environment [PROD]:
Execution (Batch/Detached/Online/Subprocess) [Batch]:
Batch queue [SYS$BATCH]:
Execute after [NOW]:
```

Version Control (cont)

Figure 7: Install Flow



Change Control



Language Specifics

File Types and Locations

- **Source files.**

  Usually found in the work directory, but if they are generated they will be found in the library directory . Typical file types include .C, .COB, .COM_SRC, .COM_APP, .H, .FOR, .SDML, .SDML_INC and .UIL Should start with application code except for special cases. If not generatyed, they should reside in the CMS library.

- **Intermediate files.**

  Usually found in the library directory. Typical file types include .LIBS, .MAP, OBJ and .OLB.

- **Tag files.**

  Usually found in the library directory. Typical file types include .TAG_CDD, .TAG_EP, .TAG_INC_1 and .TAG_INC_2. These files are used to manage dependencies which are either inefficient or impractical to manage with direct MMS dependencies.

- **Target files.**

  Usually found in the software or documentation directories. Typical file types include .COM, .EXE, .HLB, .HLP, .PS and .UID.

Language Specifics

Table 9: ACMS Application Requirements

File Type	Feature	Sematics
.ADF		ACMS application source.
	`_srv: ...` `server`	For each label ending in `_srv:` a search is made for the `server` *name* `in ...` clause. If found, is indicated a dependency upon the associated .TAG_IMG tag.
	`%include ...`	Included source. Uses a dependency on a .TAG_TRS tag.
.ADB		ACMS application database generated by the COMPILE command or the BUILD DESCRIP phase.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.
.TAG_TRS		Used to manage dependencies required by `%include ...` statements.
.TAG_IMG		Used to handle transitive dependencies to the servers used by the application.

The logical name *appl*_ACMS_REPLACE indicates whether to execute the .ADF source or use an appropriate ADU command. By default, this is assumed to have a value of false.

Language Specifics

Table 10: ACMS Menu Requirements

File Type	Feature	Sematics
.MDF		ACMS menu source.
	`menu is ...`	Names must also exist as .TDF sources and must either start with *appl*_MENUS. or start with *appl* and finish with _MNU.
	`task is ...`	Names must also exist as .TDF sources and must either start with *appl*_TASKS. or start with *appl* and finish with _TSK.
.TAG_CDD		Used to manage dependencies required by the above statements.
.MDB		ACMS menu database generated by the COMPILE command or the DESCRIP BUILD phase.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

The logical name *appl*_ACMS_REPLACE indicates whether to execute the .MDF source or use an appropriate ADU command. By default, this is assumed to have a value of false.

Language Specifics

Table 11: ACMS Task Requirements

File Type	Feature	Sematics
.TDF		ACMS task source.
	`use workspaces ..., use workspace ..., workspaces are ..., worksspace is ...`	Names must also exist as .CDDL or .DDL sources and must either start with *appl*_DICTIONARY., *appl*_FIELDS., *appl*_RECORDS. or *appl*_REPOSITORY. start with *appl* and finish with _TSK.
	`task is ..., tasks are ...`	Names must also exist as .TDF sources and must either start with *appl*_TASKS. or start with *appl* and finish with _TSK.
.TAG_CDD		Used to manage dependencies required by the above statements.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

The logical name *appl*_ACMS_REPLACE indicates whether to execute the .TDF source or use an appropriate ADU command. By default, this is assumed to have a value of false.

Language Specifics

Table 12: ACMS Task Group Requirements

File Type	Feature	Sematics
.GDF		**ACMS task group source.**
	`form[s] [{is|are}]`	**A target of the form *appl*FRM.EXE with a source of *appl*FRM.OPT and .TAG_EP dependencies for each form will be generated. The *appl*FRM.OPT file should also be generated - see .OPT file type below for details.**
	`server[s] [{is|are}]`	
	`cancel procedure`	
	`dcl process`	**A target of the form *server-name*.TAG_IMG with a source of *appl*_GRP.TDB is generated.**
	`default object file`	
	`{initial|initialization} procedure [is]`	**Generates a .TAG_EP dependency as described for `procedures are` below.**
	`procedure [server] image [is]`	
	`procedure[s] [{is|are}]`	**Generates a .TAG_EP dependency for each procedure listed.**
	`{terminal|termination} procedure [is]`	**Generates a .TAG_EP dependency as described for `procedures are` above.**
	`task[s] [{is|are}]`	**Names must also exist as .TDF sources and must either start with *appl*_TASKS. or start with *appl* and finish with _TSK.**
	`workspace[s] [{is|are}]`	**Names must also exist as .CDO sources and must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. The `with name` syntax is supported by generating a target for the alternative name based on the original name.**

Table 13: ACMS Task Group Requirements (cont)

File Type	Feature	Sematics
.TAG_CDD		Used to manage dependencies required by the above statements.
.TDB		ACMS task database generated by the COMPILE command or the BUILD DESCRIP phase.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.
.OPT		May be generated in the library directory to build a forms shareable image with a name of *appl*FRM. To use this feature, a DEVTOOLS CONVERT/GENERATE *appl*_GRP.GDF *appl*FRM.OPT command is required.

The logical name *appl*_ACMS_REPLACE indicates whether to execute the .GDF source or use an appropriate ADU command. By default, this is assumed to have a value of false.

Language Specifics

Table 14: C Requirements

File Type	Feature	Sematics
.C		**C source.**
	`#dictionary`	**Used to indicate a dependency on a CDD/Repository element. If the object is a record, its name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. If the object is a field, its name must either start with *appl*_FIELDS. or not finish with _REC.**
	`#include`	**Used to indicate a dependency on an include (i.e. .H) source.**
	`extern`	**Used with a function prototype to indicate a transitive (i.e. link time) dependency on an entry point in another module.**
	`main`	**The presence of a `main` function indicates that this source forms the root module for an image. As a result, no dependency is generated to insert the resulting object module into the object library.**
.H		**C include source.**
	`#dictionary`	**See C source.**
	`#include`	**See C source.**
	`extern`	**See C source**
.TAG_CDD		**Used to manage dependencies required by `#dictionary` statements.**
.TAG_INC_1		**Used to manage dependencies caused by nested `#include` statements.**
.TAG_INC_2		**Used to manage transitive dependencies caused by `extern` statements in `#include` sources.**
.LIS, .OBJ		**Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.**
***appl*OBJ.OLB**		**Created in the library directory to store the resultant object module. Note that this module is not inserted if the .C source contained a `main` function.**

Table 14 (Cont.): C Requirements

File Type	Feature	Sematics
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 15: C++ Requirements

File Type	Feature	Sematics
.CXX		**C++ source.**
	`#dictionary`	**Used to indicate a dependency on a CDD/Repository element. If the object is a record, its name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. If the object is a field, its name must either start with *appl*_FIELDS. or not finish with _REC.**
	`#include`	**Used to indicate a dependency on an include (i.e. .HXX and .IXX) source.**
	`extern`	**Used with a function prototype to indicate a transitive (i.e. link time) dependency on an entry point in another module.**
	`main`	**The presence of a `main` function indicates that this source forms the root module for an image. As a result, no dependency is generated to insert the resulting object module into the object library.**
.HXX, .IXX		**C++ include source.**
	`#dictionary`	**See C++ source.**
	`#include`	**See C++ source.**
	`extern`	**See C++ source**
.TAG_CDD		**Used to manage dependencies required by `#dictionary` statements.**
.TAG_INC_1		**Used to manage dependencies caused by nested `#include` statements.**
.TAG_INC_2		**Used to manage transitive dependencies caused by `extern` statements in `#include` sources.**
.LIS, .OBJ		**Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.**
***appl*OBJ.OLB**		**Created in the library directory to store the resultant object module. Note that this module is not inserted if the .C source contained a `main` function.**

Table 15 (Cont.): C++ Requirements

File Type	Feature	Sematics
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 16: CDD/Repository Requirements

File Name Suffix and File Type	Feature	Sematics
_FLD.CDO		CDD/Repository field source.
	`define field`	Global field definition. Field names should either start with *appl*_FIELDS. or not finish with _REC.
_REC.CDO		CDD/Repository record source.
	`define field`	Local field definition. Same rules apply as for global fields.
	`define record`	Record definition. The name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 17: CDD Requirements

File Name Suffix and File Type	Feature	Sematics
_REC.CDDL, _REC.DDL		CDD record source.
	`define record`	Record definition. The name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 18: Cobol Requirements

File Type	Feature	Sematics
.COB		Cobol source.
	filename	Must start with application code. Should reside in the CMS library.
	`program-id`	Must be the same as the filename.
	`copy ...`	Copy in a source file. The file should reside in the work directory.
	`copy ... from dictionary`	Used to indicate a dependency on a CDD/Repository element. If the object is a record, its name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. If the object is a field, its name must either start with *appl*_FIELDS. or not finish with _REC.
	`copy ... in ...`	Copy a source from a library. The library should reside in the library directory. It should have a file name of *appl*TXT.TLB.
	`call`	Used to create a transitive dependency via .TAG_EP. Called program name must be in quotes.
.TAG_CDD		Used to manage dependencies required by `copy ... from dictionary` statements.
.TAG_CPY_1		Used to manage dependencies caused by nested `copy ...` and `copy ... in ...` statements.
.TAG_CPY_2		Used to manage transitive dependencies caused by `call` statements in copy files and libraries.
.LIS, .OBJ		Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.
*appl*OBJ.OLB		Created in the library directory to store the resultant object module.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 19: DECforms Requirements

File Type	Feature	Sematics
.IFDL		DECforms source.
	`form`	Must be the same as the filename. Should start with the application code and finish with _FRM.
	`copy ...`	Copy in a source file. The file should reside in the work directory.
	`copy ... from dictionary`	Used to indicate a dependency on a CDD/Repository element. If the object is a record, its name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. If the object is a field, its name must either start with *appl*_FIELDS. or not finish with _REC.
	`copy ... in ...`	Copy a source from a library. The library should reside in the library directory. It should have a file name of *appl*TXT.TLB.
	`call`	Used to create a transitive dependency via .TAG_EP. Called program name must be in quotes.
.TAG_CDD		Used to manage dependencies required by `copy ... from dictionary` statements.
.TAG_CPY_1		Used to manage dependencies caused by nested `copy ...` and `copy ... in ...` statements.
.TAG_CPY_2		Used to manage transitive dependencies caused by `call` statements in copy files and libraries.
.FORM		DECforms intermediate binary format.
.LIS, .OBJ		Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.
*appl*OBJ.OLB		Created in the library directory to store the resultant object module.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 20: Fortran Requirements

File Type	Feature	Sematics
.FOR		Fortran source.
.LIS, .OBJ		Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.
*appl*OBJ.OLB		Created in the library directory to store the resultant object module.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 21: Oracle Rdb Requirements

File Name Suffix and File Type	Feature	Sematics
_CK*n*.SQL		Check constraint definition.
_DOM.SQL		Domain definition.
_FK*n*.SQL		Foreign key constraint definition.
_MAP.SQL		Storage map definition.
_PK.SQL		Primary key constraint definition.
_PIDX.SQL		Primary index definition.
_RTN		Routine definition.
_SIDX*n*		Secondary (alternative) index definition.
_TBL.SQL		Table definition.
_TRG.SQL _TRGR.SQL		Trigger definition.
_VIEW.SQL _VW.SQL		View definition.
.SQLMOD		Module source.
	`procedure`	Used to create a transitive dependency via .TAG_EP.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

The following SQL include lists are generated in the application architecture independent library directory during the BUILD SCAN phase:

- *appl*_CONSTRAINTS.SQL
- *appl*_DOMAINS.SQL
- *appl*_INDICES.SQL

- *appl*_ROUTINES.SQL
- *appl*_STORAGE_MAPS.SQL
- *appl*_TABLES.SQL
- *appl*_TRIGGERS.SQL
- *appl*_VIEWS.SQL

The following MMS dependency list is generated in the application architecture independent library directory during the BUILD SCAN phase:

- *appl*_SCHEMA.MMS

Language Specifics

Table 22: TDMS Forms Requirements

File Type	Feature	Sematics
.FRM_TDMS		CDD backup of TDMS form used as a source.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 23: TDMS Request Requirements

File Type	Feature	Sematics
.RDF		TDMS Request source.
	`form is ..., forms are ...`	Names must also exist as .FRM_TDMS sources and must either start with *appl*_FORMS. or start with *appl* and finish with _FRM.
	`record is ..., records are ...`	Used to indicate a dependency on a CDD element. If the object is a record, its name must either start with *appl*_RECORDS. or start with *appl* and finish with _REC. If the object is a field, its name must either start with *appl*_FIELDS. or not finish with _REC.
.TAG_CDD		Used to manage dependencies required by the above statements.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

The logical name *appl*_TDMS_REPLACE indicates whether to execute the .RDF source or use an appropriate RDU command. By default, this is assumed to have a value of false.

Language Specifics

Table 24: TDMS Request Library Requirements

File Type	Feature	Sematics
.LDF		TDMS request library source.
	`request is ..., requests are ...`	Used to indicate a dependency on a TDMS request definition. Its name must either start with *appl_* REQUESTS. or start with *appl* and finish with _REQ.
.TAG_CDD		Used to manage dependencies required by the above statements.
.RLB		Generated in the software directory by the COMPILE command or the BUILD DESCRIP phase.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

The logical name *appl*_TDMS_REPLACE indicates whether to execute the .LDF source or use an appropriate RDU command. By default, this is assumed to have a value of false.

Language Specifics

Table 25: DECdocument Requirements

File Type	Feature	Sematics
.SDML		DECdocument source. See Table 26 for file name suffix conventions.
	<CONTENTS_FILE>	Include /CONTENTS in the generated DOCUMENT command. This tag must only be used in a .SDML source - it is not recognised in a .SDML_INC include source.
	<EXAMPLE_FILE>, <FIGURE_FILE>, <INCLUDE>	Generate a dependency on the specified include file.
	<INDEX_FILE>	Include /INDEX in the generated DOCUMENT command. This tag must only be used in a .SDML source - it is not recognised in a .SDML_INC include source.
.SDML_INC		DECdocument include source.
	<EXAMPLE_FILE>, <FIGURE_FILE>, <INCLUDE>	See .SDML source.
.GRA		DECdocument graphic source.
.INX_LIST		Used during the creation of a master index.
.COM, .EPS, .HLP, .INT_TOC, .INX, .LIS, .MAP_LIS, .TAG_INX, .TAG_SDML, .TEX		Generated in the library directory by the COMPILE command or the BUILD DESCRIP phase.
.DECW$BOOK, .GIF, .HTML, .PS, .TXT		Generated in the documentation directory by the COMPILE command or the BUILD DESCRIP phase.
.MMS_INC		Generated in the library directory by the BUILD RULES phase.

Language Specifics

Table 26: DECdocument File Name Suffices

File Name Siffix	Usage
_CVR	Cover note source for generating PostScript and text output.
_GDE	Guide source for generating bookreader, master index and PostScript output.
_HLP	Help source.
_IDX	Master index sources for generating PostScript output. Two sources must exists, one with a file type of .SMDL and the other of a file type of .INX_LIST.
_INS	Guide source for generating bookreader, PostScript and text output.
_OVH	Overhead source for generating PostScript output.
RELEASE NOTES	Release notes source for generating bookreader, PostScript and .RELEASE_NOTES output.
_REF	Reference manual source for generating bookreader, master index and PostScript output.