

Software Product Description

PRODUCT NAME: SysWorks™ Developer V3.5

SPD SWRK-DEV-35

DESCRIPTION

SysWorks™ Developer provides a framework which promotes efficient development and effective version control. This is achieved by maintaining source libraries and assisting in developing, maintaining, building, testing and migrating applications. The framework includes conventions, use of Digital and third party layered products and SysWorks™ commands and menus.

Languages and Products

SysWorks™ Developer supports application development using the following languages and products:

- ACA Services
- ACMS
- Basic
- C
- C++
- CDD, CDD/Plus, CDD/Repository
- Cobol
- DCL
- DECdocument
- DECforms
- DECset (CMS, DTM, LSE, MMS, PCA, SCA)
- DECwindows UIL
- FMS
- Fortran
- Macro-32
- ObjectBroker
- Pascal
- Rdb/VMS, DEC Rdb, Oracle Rdb
- RDML Precompilers, RDO, RDO Precompilers, SQL, SQL Module, SQL Precompilers

- Runoff
- TDMS

Environments

SysWorks™ Developer supports logical separation of application environments such as development, testing and production.

An environment prime model is used. In this model the environment is superior to the application in the context hierarchy. This model was chosen as it reflects the need of organizations developing a number of applications on the same VMSccluster or which are interrelated. The model is equally suitable for smaller organizations or where a VMSccluster is used to develop a single application.

Environment Types

Initially there are six environment types, these being:

- Common - this typically contains the CMS and DTM libraries - example codes include APPL.
- Development - typically uses the mainline in CMS - example codes include DEV, DEVL and DEVT.
- Development Testing - testing for mainline versions - example codes include DTST.
- Maintenance - typically uses a variant line in CMS - example codes include MNT and MNTN.
- Maintenance Testing - testing for variant versions - example codes include MTST.
- Production - example codes include PROD and TRNG.

Initially there are two application environments, one common called APPL and one production called PROD.

Common vs Specific Scope

When a developer sets their context to an application environment, their scope can be set such that all changes are visible after compilation to all other developers. This is known as the *common* scope. When a source is compiled or a target built, the results are placed in the application common directory structures.

Alternatively, changes to sources are not visible until an explicit promotion back to the common scope is made. This is known as the *specific* scope. When a source is compiled or a target built, the results are placed in developer specific directories. Note that the specific scope does not work with ACMS.

One of the attributes of an environment type is whether it supports these two scopes. Some environment types such as production only support a common scope.

Multi-variant Development

SysWorks™ Developer supports parallel development of an application in environment variants. These are represented by a separate sub-directory structure to the main environment in a manner similar to specific versus common scopes. Merging of sources from a variant back into the mainline can wait until it is known which sources are required for a release.

Multi-version Maintenance

SysWorks™ Developer supports the maintenance of multiple versions of an application. The most recent version occupies the main environment directory structure. Older versions are placed in sub-directory structures within the environment. Unlike multi-variant development, each version is maintained independently.

Selecting a Context

Developers can move between application environments, scope, variants, versions and their home (i.e. login) areas using the CONTEXT command which has a number of subcommands and qualifiers to indicate the desired context change.

Each move may involve the use of DCL command procedures to define context specific information. These include the following:

- ENTER.COM Define context specific symbol values and process or job logical names. Indicate dependencies on other applications such as a utility or corporate database application.
- EXIT.COM Delete context specific symbols and process or job logical names.
- LOGICALS.COM Define context specific shared logical names.

Directories

SysWorks™ Developer uses standard directories including:

- Dictionary - CDD/Plus or CDD/Repository anchor directory.
- Documentation - resulting manuals, help libraries etc.
- Kit - where savesets are built for installable software. This directory is normally provided in testing and common environments.
- Journal - where Rdb/VMS and RMS journals are placed.
- Library - intermediate build files such as objects, lists, maps etc. - also includes generated MMS scripts. This directory is not normally provided in production environments.
- Runtime - where log files, etc are placed.
- Software - resulting target images, DCL command procedures etc.
- Source - includes CMS and DTM libraries. This directory is normally provided in common environments.
- Test - resulting target images, DCL command procedures etc. which are used solely for testing.
- Work - intermediate area for extraction from CMS library by MMS. This directory is not normally provided in production environments.

Other features include:

- Permits site or application specific changes to the default directory names and logical names.
- Directories can be placed on separate disk volumes.
- A directory can exist on multiple disk volumes when access to the directory is for files. An example of this is the data directory. It is not possible for build directories such as the library and work directories to have multiple instances.

Logical Names

SysWorks™ Developer uses application logical name tables. These may be:

- System wide and created at system boot time by either the SysWorks™ Administrator product or site specific procedures;
- Process wide and created by the CONTEXT command.

They are also:

- Normally used to access the application directory structure. If a directory is moved, the applications LOGICALS.COM can be re-executed to update the logical name equivalences.
- Used to indicate default or preferred actions for a number of tools. For example the logical name *appl_ACMS_REPLACE* may contain a value which indicates whether an ACMS source contains the ADU REPLACE command and SysWorks™ Developer should simply execute (i.e. DCL @) the source or whether it should generate an ADU REPLACE *source-file-spec* command.

Source Control Features

SysWorks™ Developer includes the following features for managing sources:

- Enhances CMS source library control by integrating it with multiple environments and versions. Extensions include:
 - Both single and multiple CMS library models are supported.
 - Use of CMS classes for environments and versions. Individual application environments can nominate mainline or variant classes.
 - Use of CMS class search lists. This is used to enable the specific scope and variant development environments.
 - The meaning of the CMS commands changes depending upon the class and variant requirements. For example, a DELETE ELEMENT command would effectively become a REMOVE GENERATION command if a CMS class was being used.
- Provides reports on unsound organization of classes within a CMS library including unused (i.e. not in any class) elements.
- Integrates the CMS environment and application logical name tables with LSE and DECwindows Motif when LSE is selected from a view of the developer's application work directory.

- Provides a consistent interface with EDT, EVE and LSE between direct, sub-process and DECwindows Motif invocation of these editors. In sub-process mode, the normal direct command is replaced in a way which makes the kept sub-process almost invisible to the developer.

Compiling and Building

SysWorks™ Developer includes the following features for compiling sources and building applications:

- Provides a multi-phase BUILD command integrated with DECset products such as CMS, MMS and SCA. Multiple phases are required to allow the use of MMS to generate the final MMS script used to build the application.
- Also provides a COMPILE command to compile, and optionally link, any supported source or collection of sources. This command interacts with the BUILD command to reduce the need to re-compile sources during builds.
- Provides MMS generators for most OpenVMS languages and layered products. These MMS generators produce separate compile and build time dependencies. This removes the requirement of most other MMS generators to parse all sources required to build a given target in order to generate a script to build that target.
- Includes extra generators for some layered products. For example, a task group generator is provided for use with ACMS.
- Uses tag files for CDD element dependencies and a kept sub-process for loading new or updated CDD sources in order to significantly improve the performance of building applications which use the CDD including those which use ACMS and TDMS.

Version Control Features

SysWorks™ Developer provides a comprehensive version control menu which includes the following tasks:

- BUILD - Build the application. This is equivalent to the BUILD command.
- COMPARE - Compare the CMS library against the work directory and produce a DCL command procedure to update the CMS library if necessary.
- CLEANUP - Compares the library directory against the work directory and CMS library, and produces a DCL command procedure to remove unused sources, and tidy the disk directories.

- TRIAL - Migrate a development or maintenance style (eg. DEV or MNT) environment to a development or maintenance testing style environment (eg. DTST or MTST).
- DEVELOP - Take a released source version and develop a new version from it in a development environment.
- MAINTAIN - Take a released source version and perform maintenance on it in a maintenance environment.
- SPLIT - Equivalent to a DEVELOP and a MAINTAIN together.

Miscellaneous

Other features include:

- Restricts access to SysWorks™ Developer tools to members of the DEVELOPER system user class when used with the SysWorks™ Administrator User Management Sub-System.
- Adds views to a developers FileView and Session Manager menus for each application environment. These views are added by SysWorks™ Administrator User Management Sub-System or by developers using a SysWorks™ command procedure.
- Provides DECdocument designs for standard OpenVMS style manuals.

SOFTWARE REQUIREMENTS

Where software versions are supplied, they indicate a minimum version number. Where multiple versions are supplied, these indicate versions of the product which have been tested with SysWorks.

Prerequisite Software

The following software is required by SysWorks™ Developer:

- OpenVMS for Alpha or VAX V7.2
- DECnet V5.5 or DECnet/OSI V6.1
- DECset V12B
 - CMS
 - MMS

Optional Software

The following software will be actively used by SysWorks™ Developer when present:

- DECset V12B
 - DTM
 - LSE
 - SCA
 - PCA
- DECwindows/Motif V1.2-3

Supported Software

The following software is supported by SysWorks™ when present:

- 20/20
- ACA Services
- ACMS V3.2
- All-In-1 V2.4
- All-In-1 Shared File Cabinet
- CDD/Plus V4.x, CDD/Repository V5.x
- Data Distributor
- Datatrieve
- DEC Ada
- DEC C
- DEC C++
- DEC DB Tools
 - RdbExpert
 - Graphical Schema Editor
 - InstantSQL
- DEC Fortran
- DEC Notes
- DEC Pascal
- DEC TCP/IP Services
- DECdecision
- DECdocument
- DECfax
- DECforms V1.3
- DECgraph
- DECquery
- DECset V12B
- DECtrace
- DECwrite V3.0

- Macro-32
- Mailworks
- ObjectBorker
- Pathworks
 - LAN Manager V5.0C
 - Macintosh V1.3
 - Netware V1.0
- POLYCENTRE
 - Accounting Chargeback
 - Capacity Planner
 - File Optimizer
 - Framework
 - Network Manager
 - Performance Advisor
 - Performance Data Collector
 - Scheduler
- Rally
- Rdb/VMS V4.0, V4.2, DEC Rdb V5.1, V6.0, Oracle Rdb V6.1
 - RDML Precompilers
 - RDO
 - RDO Precompilers
 - SQL
 - SQL Module
 - SQL Precompilers
 - SQL Services
- Storage Library System
- SoftPC V4.0
- TeamLinks
 - Macintosh
 - Windows
- TeamRoute for OpenVMS
- VAX Basic
- VAX Cobol
- VAX FMS
- VAX Message Router
 - DDS
- VAX TDMS V1.9A

- WordPerfect V5.0, V5.1

DOCUMENTATION

All SysWorks™ software is supplied with comprehensive and consistent online help and Bookreader and PostScript manuals.

SOFTWARE LICENSING

This software is furnished under the licensing provisions of Corpita Pty Ltd's Standard Terms and Conditions.

SOFTWARE WARRANTY

Warranty for this software product is provided by Corpita Pty Ltd with the purchase of a license for the product.

SERVICES

A variety of services are offered with SysWorks™ software. These include:

- Installation and configuration;
- Adaption to existing site standards;
- Establishment of application baselines;
- Addition of Digital or third party software to SysWorks™ Developer and/or SysWorks™ Administrator startup;

NOTES

- ™ SysWorks™ is a registered trademark of Corpita Pty Ltd.
- ™ 20/20 is a trademark of Access Technologies Inc.
- ™ ACMS, All-In-1, All-In-1 Mail, AlphaServer, AlphaStation, CDD, CDD/Plus, CMS, DCL, DECdecision, DECforms, DECgraph, DECquery, DECset, DECwindows, DECwrite, DTM, LSE, MMS, OpenVMS, PathWorks, PCA, Rdb/VMS, SCA, TDMS, VAX, VAXcluster, VAXset and VMScluster are all trademarks of or copyright by Digital Equipment Corporation.
- ® Macintosh is a registered trademark of Apple Computer Inc.
- © Motif is copyright by the Open Software Foundation.
- ® PostScript is a registered trademark of Adobe Corp.
- ® SoftPC is a registered trademark of Insignia Solutions.
- ® UNIX is a registered trademark of Novell Inc.
- ® WordPerfect is a registered trademark of WPCorp.