

Software Product Information

PRODUCT NAME: DSM for OpenVMS Version 6.5

Part Number: IS-DSMIA-VM

This Software Product Information sheet describes two products: DSM for OpenVMS Alpha and DSM for OpenVMS VAX (DSM).

DSM for OpenVMS™ is an implementation of the ANSI Standard Specification for M (X11.1-1990) that is layered on the OpenVMS™ operating system. It is a high-level programming language and a multiuser data management system in which many users can share a common database. DSM software supports a superset of the ANSI M specification and fully uses the capabilities of the OpenVMS operating system.

DSM Language

The DSM language couples the flexible processing of variable length string data with a high-performance database system, making interactive database application systems easier to implement and maintain.

The DSM language includes the following extensions to the ANSI Standard M Specification:

- A 512-character maximum string length for global variables and a 32,767-character maximum string length for local variables.
- A 245-character maximum string length for sub-scripted local and global variable names.
- Commands and functions to load, save, and edit DSM routines.
- Language elements that implement a subset of the OpenVMS I/O options for the support of terminals, including foreign terminals, magnetic tape devices, mailboxes, and OpenVMS RMS sequential, relative, and indexed sequential files. The use of DECnet, DEC TCP/IP Services for OpenVMS (UCX), Wollongong WIN/TCP, and TCPware™ for OpenVMS for remote file access and task-to-task communication is also possible.
- An I/O command interface to DEC TCP/IP Services for OpenVMS (UCX), Wollongong WIN/TCP, and TCPware™ for OpenVMS.. This allows application software to access TCP/IP network devices using

M OPEN, USE, READ, WRITE, and CLOSE commands.

- An Interface to DECwindows System and DECimage Application Services software.
- A maximum routine line length of 512 characters.
- MDC Type A Language Extensions.
- Routine source code protection.
- Flexible device control mnemonic space definition and management for use with the M WRITE / and READ / syntax.
- Visual M™ remote application server support.
- DEC DB Integrator Gateway for DSM™ support for Open M/SQL™.

In addition to supporting the calling of user-written procedures written in other programming languages, DSM software includes built-in interfaces to selected OpenVMS system services and run-time library routines. This gives the DSM language the ability to mount and dis-mount disk or magnetic tape volumes, spawn DCL subprocesses, manipulate logical names, and call mathematical and text manipulation functions, or editors.

For high run-time efficiency, the DSM language is implemented as a pre-compiler that generates intermediate code. During the precompilation process, which is invisible to the user, DSM software performs operations that optimize subsequent run-time execution. For example, precompilation strips comments, checks syntax, optimizes label references, and transforms numeric constants into an internal representation.

Global Variables

The DSM language allows symbolic reference to hierarchical arrays called global variables (or simply globals) that can be accessed concurrently by many users of a DSM application. DSM software provides a high-performance implementation of global variables using multiway trees.

DSM for OpenVMS Version 6.5

DSM stores global variables and DSM routines on data structures called DSM volume sets. Each is comprised of up to eight logical volumes. Volumes in a DSM volume set can be either files on an OpenVMS volume or disk packs mounted as foreign OpenVMS volumes. In a DSM run-time environment, up to 31 volume sets can be present simultaneously.

Through translation tables, you can map portions of globals across any number of volume sets. You can change and update the translation table mappings without bringing down the configuration to which they apply. You can also use the translation tables for more than one configuration.

Database Sharing in a VMScluster Environment

In OpenVMS configurations that support a VMScluster™ environment, the DSM for OpenVMS software allows volume sets to be mounted and made accessible to all members of a VMScluster environment.

The globals and routines resident within the cluster-mounted volume sets are accessible in the same manner as a locally mounted volume set. Each cluster member maintains and updates a memory disk buffer cache for the cluster-mounted volume set. VMScluster software coordinates the access and update of physical disk blocks.

DSM configurations can also be run in heterogeneous VMScluster environments consisting of VAX and Alpha CPUs.

M Windowing Application Programming Interface

The DSM product includes an implementation of the standard M Windowing Application Programming Interface (MWAPI). MWAPI provides a high level and portable method of creating a graphical user interface (GUI) for DSM applications.

Distributed Data Processing

DSM applications can access globals either locally or on a remote system. The DSM software supports remote access to globals through a DSM-specific Distributed Data Processing (DDP) protocol.

Distributed global access can be done either explicitly, through the extended DSM global syntax, or transparently to DSM applications through a system of global translation. Optionally, the DSM environment manager can specify the replication of one or more globals across several systems.

DSM software implements this high-performance protocol on Ethernet in a manner compatible with DSM for DEC OSF/1 AXP (SPD 47.60.xx). (The operating system is now called Digital UNIX.)

This allows the user to set up DSM networks consisting of any combination of Alpha processors, DECstation or DECsystem processors, and VAX processors. The DDP autoconfiguration facility automatically updates the software tables of members of a DDP network when a new system goes on or off line.

DSM software also supports DDP over DECnet Phase IV logical links. This allows remote global access between VAX and Alpha systems in a wide area network. DDP over DECnet Phase V networks is not supported.

In addition, DSM supports DDP over DEC TCP/IP Services for OpenVMS (UCX) and TCPware™ for OpenVMS.

DSM allows users to modify all DDP communications characteristics (except for the DDP node name) in a running DSM configuration. Modifications users make to DDP communications characteristics become active after they stop and start DDP. Users do not have to shut down and restart the configuration.

DEC DB Integrator Gateway for DSM

The DEC DB Integrator Gateway is a general-purpose database tool that supplies access and data integration among a wide variety of relational databases. The DEC DB Integrator Gateway for DSM supplies relational access to a DSM database. In DEC DB Integrator terms, DSM supplies a non-SQL data service (NSDS) drive as the interface between the DB Integrator and the DSM database.

Previously, DSM supported data-dictionary specific M data and metadata drivers for VA Fileman and DASL and also allowed users to create their own M routine drivers for custom data dictionaries. DSM for OpenVMS includes an M routine driver, %MSQL, that provides a read/insert/update/delete interface for InterSystems' Open M/SQL. This new Open M/SQL interface is compatible with Digital Equipment Corporation's NSDS Version 3.0C and later.

After-Image Journaling

DSM supports a journaling facility to provide a record on secondary storage of all operations that modify the database (SET and KILL of global variables). Journaling can be done on a global-by-global basis or for a whole DSM environment. In the event of database degradation, it is possible to restore the current database from a full backup and the journal files using a dejournaling utility.

Before-Image Journaling

DSM software provides a journaling capability that records the physical image of a database block before modifications have been made. This facility preserves database integrity in the event of a CPU failure. The physical before-image journal contains only a very small number of blocks and is used to quickly roll back any operations in progress at the time of the failure. The operations protected are SETs to globals that result in a database block split, or KILLS that result in a database block merge.

Transaction Processing

DSM provides extensions to the ANSI M language that allow programmers to group global accesses into well-defined transaction recovery units. The use of transaction recovery units gives application designers the ability to create highly reliable and efficient fault-tolerant application systems.

Database updates made within the context of a recovery unit are treated as a single unit of work. On conclusion of a recovery unit, DSM ensures that updates are either committed entirely or not at all. If a failure occurs during recovery unit processing, any partial updates made to the database are rolled back so that the database is restored to its original state before the recovery unit began.

Through the conventional use of the M LOCK command and the application of the recovery unit commands, programmers can construct database transactions that exhibit the transaction processing properties of atomicity, consistency, isolation, and durability (ACID properties).

DSM supports rollback of recovery units through a system of before-image and after-image journal files. Transaction recovery unit processing is supported on database volume sets that are mounted locally. DSM does not support transaction processing on volume sets mounted cluster-wide across multiple nodes within a VMScluster environment.

Recovery of updates made to databases that are shared using DSM Distributed Data Processing (DDP) is not supported within a transaction recovery unit.

Mountable Database Sets

DSM allows you to mount and dismount individual database sets while a configuration is still running. This facility reduces system down time for activities such as extending volume sets, adding volumes to a volume set, or adding new applications and volume sets to a running configuration.

DECwindows System Interface

DSM provides a complete binding for the DSM language to the DECwindows system. The binding is implemented as three separate external call packages, which provide an M interface to the XTOOLKIT, XMO-TIF, and XMOTIF Resource Manager program libraries. The binding conforms to the proposed ANSI M Standard Specification.

An additional package provides data manipulation primitives for allocating and modifying X Window data structures. These packages give the programmer complete access to X Window procedures for creating sophisticated graphical user interfaces.

A set of sample M programs is provided with complete documentation of the interface.

Incremental Backup/Restore

DSM provides an Incremental Backup Utility (^BACKUP) for use with the DSM database. The ^BACKUP Utility backs up only those blocks that have been modified since the last backup. Because modified blocks usually constitute a small portion of an entire DSM database, the ^BACKUP Utility represents a major reduction in the time required for backup and the amount of backup storage required.

The ^BACKUP Utility implements a multiphase backup. The initial phases allow database read and write operations to occur unrestricted on both locally mounted and cluster-mounted volume sets. The last phase temporarily inhibits database operations for a short period to establish a final consistency check point.

Full database backup of DSM for OpenVMS volume set files is accomplished using the OpenVMS Backup utility. Restoration of an incremental backup is accomplished using the Incremental Restore Utility to apply an incremental backup file onto a fully restored primary database. The backup media supported by the ^BACKUP Utility is OpenVMS sequential disk files.

Callable Routines Interface

The callable routines interface allows routines written in software languages that run under the OpenVMS operating system to directly call the DSM shareable image. This capability gives programs written in languages such as FORTRAN, C, PASCAL, and others complete read and write access to globals resident in a DSM volume set. Functions such as global set, kill, get, or lock to a DSM database can be accomplished using many OpenVMS layered languages. Also, external programs can declare recovery units, access the DSM local symbol table, and execute M code.

DSM for OpenVMS Version 6.5

The following functions are available to external programs:

- Callable interface initialization and rundown
- Global database access
- Declare recovery unit (start, commit, abort)
- LOCK command
- Local symbol access
- M command and routine execution

Visual M Remote Application Server Support

DSM for OpenVMS can function as a remote application server to InterSystems' Visual M™. Visual M is a graphical M interface running on any of the following Microsoft Windows Platforms:

- Windows 3.1™
- Windows NT™
- Windows 95™

Using Visual M, you can integrate Visual Basic™ programs with DSM applications operating in a DSM for OpenVMS environment to create distributed client-server applications.

Utilities

The DSM product provides application and system level utilities written in the DSM language. Application level utilities help the DSM programmer develop and maintain application software and data. For example, there are utilities to edit and display DSM routines and globals.

System level utilities allow the management of a DSM application environment. The system manager can create and extend DSM volume sets, specify DSM configuration parameters such as access control (security), the number of concurrent DSM jobs and disk buffers, define DSM command line defaults, establish DDP links and control journaling. For increased efficiency of routine calling, the system manager can map a selected set of DSM application routines in a shared virtual memory section, resulting in higher application throughput.

Include with the system level utilities is a new performance monitoring facility. This facility, ^PMF, is a complete set of performance monitoring tools that can summarize DSM performance metrics across multiple nodes of a VMScluster.

DSM Application Software Library (DASL)

DSM includes the DSM Application Software Library (DASL) software. The DASL software is an application creation tool that combines a Data Dictionary, Screen (form) Compiler, Report Compiler, and an end-user SQL-compliant Query Driver. DASL software uses code generation techniques to transform DASL commands and SQL statements into efficient M code.

DASL software is a set of menu-driven tools designed for developers of interactive applications. The DASL package automates many of the tasks involved in defining and documenting a database: generating data input, menu, and help text screens, designing reports for data output, and creating tables for database queries.

The DASL package is fully integrated with DSM. Therefore, DASL software provides both 4GL and 3GL capabilities within a single environment. Users interact with DASL software through menus and screens that were developed using DASL software and are stored as DSM routines.

The DASL package includes six major modules, each representing a group of related functions that programmers use to develop applications.

- The Data Dictionary defines and documents the application database.
- The Screen Driver contains options to develop and modify screen displays for data entry, menus, and help text.
- The Report Driver contains options to define and modify application reports.
- The Query Driver contains options to define tables for the DASL database and create queries to extract data from the tables. The DASL Query Driver conforms to the syntax of the ANSI Standard Database Language SQL (Structured Query Language).
- The Development Environment performs a variety of development tasks such as specifying application parameters, or creating an international application using the DASL Language Utilities.
- The Application Environment provides facilities for controlling the application's run-time environment such as a security system, message center, query database, and error logging.

Source Code Information

DSM provides example files for writing external routines and external package tables in the SYS\$EXAMPLES directory. Examples of programs using the callable routines interface are also supplied. This source code is provided on an as is basis without any warranty of any kind, either expressed or implied.

DSM for OpenVMS Version 6.5

Hardware Requirements

Processors Supported:

For Alpha Systems:

Alpha:

- DEC pc 150 AXP Server
- DEC 2000 Models 300/500 Server
- DEC 3000 Model 300 Server
- DEC 3000 Model 300L System
- DEC 3000 Model 300X System
- DEC 3000 Model 400 Workstation
- DEC 3000 Model 400 S Server
- DEC 3000 Model 500 Workstation
- DEC 3000 Model 500S Server
- DEC 3000 Model 500X Workstation
- DEC 3000 Model 600 Workstation
- DEC 3000 Model 600S Server
- DEC 3000 Model 800 Workstation
- DEC 3000 Model 800S Server
- DEC 4000 Model 600 System
- DEC 4000 Model 700 System
- AlphaServer 8400 5/300
- DEC 7000 Model 600 System
- DEC 10000 Model 600 System
- AlphaServer 8200 5/300
- Digital 2100 Server Model A500MP
- Digital 2100 Server Model A600MP
- AlphaServer 400 4/166
- AlphaServer 1000 4/200
- AlphaServer 2000 4/233
- AlphaServer 2100 4/233
- AlphaStation 400 4/166
- AlphaStation 400 4/266
- AlphaStation 250 4/233
- AlphaStation 200 4/166
- AlphaStation 200 4/100

For VAX Systems:

VAX:

- VAXft Model 110
- VAXft Model 310
- VAXft Model 410
- VAXft Model 610

VAXft Model 612

VAX 4000 Model 100
VAX 4000 Model 200
VAX 4000 Model 300
VAX 4000 Model 400
VAX 4000 Model 500
VAX 4000 Model 600

VAX 6000 Model 200 Series
VAX 6000 Model 300 Series
VAX 6000 Model 400 Series
VAX 6000 Model 500 Series
VAX 6000 Model 600 Series

VAX 7000 Model 600 Series

VAX 8200, VAX 8250, VAX 8300
VAX 8350, VAX 8500, VAX 8530
VAX 8550, VAX 8600, VAX 8650
VAX 8700, VAX 8800, VAX 8810
VAX 8820, VAX 8830, VAX 8840

VAX 9000 Model 110
VAX 9000 Model 210
VAX 9000 Model 300 Series
VAX 9000 Model 400 Series

VAX 10000 Model 600 Series

VAX 9000 Model 400 Series

VAX-11/730, VAX-11/750
VAX-11/780, VAX-11/785

MicroVAX:

MicroVAX II, MicroVAX 2000
MicroVAX 3100 Model 10/10E
MicroVAX 3100 Model 20/20E
MicroVAX 3100 Model 30
MicroVAX 3100 Model 40
MicroVAX 3100 Model 80
MicroVAX 3100 Model 90
MicroVAX 3300, MicroVAX 3400
MicroVAX 3500, MicroVAX 3600

DSM for OpenVMS Version 6.5

MicroVAX 3800, MicroVAX 3900

VAXstation: VAXstation II, VAXstation 2000
 VAXstation 3100 Model 30
 VAXstation 3100 Model 38
 VAXstation 3100 Model 40
 VAXstation 3100 Model 48
 VAXstation 3100 Model 76
 VAXstation 3200, VAXstation 3500
 VAXstation 3520, VAXstation 3540

VAXstation 4000 Model 60
 VAXstation 4000 Model 90
 VAXstation 4000 VLC

VAXserver: VAXserver 3100 Model 10/10E
 VAXserver 3100 Model 20/20E
 VAXserver 3300, VAXserver 3400,
 VAXserver 3500
 VAXserver 3600, VAXserver 3602,
 VAXserver 3800
 VAXserver 3900

VAXserver 4000 Model 200
 VAXserver 4000 Model 300
 VAXserver 4000 Model 500

VAXserver 6000 Model 210
 VAXserver 6000 Model 220
 VAXserver 6000 Model 310
 VAXserver 6000 Model 320
 VAXserver 6000 Model 410
 VAXserver 6000 Model 420
 VAXserver 6000 Model 510
 VAXserver 6000 Model 520
 VAXserver 6000 Model 610
 VAXserver 6000 Model 620
 VAXserver 6000 Model 630

Processors Not Supported:

MicroVAX I VAXstation I
 VAX-11/725 VAX-11/782
 VAXstation 8000

Processor Restrictions:

- A TK50 Tape Drive is required for standalone MicroVAX 2000 and VAXstation 2000 systems.

- A system configuration must have memory of 2MB of physical storage.

Disk Space Requirements (Block Cluster Size = 1):

For Alpha Systems:

DSM only:

For installation: 34,500 blocks

For use (permanent): 23,500 blocks

With DASL only:

For installation: 57,000 blocks

For use (permanent): 37,000 blocks

With M WAPI and Windows only:

For installation: 38,500 blocks

For use (permanent): 25,500 blocks

With DASL, M WAPI and Windows:

For installation: 60,500 blocks

For use (permanent): 39,000 blocks

For VAX Systems:

DSM only:

For installation: 25,100 blocks

For use (permanent): 22,000 blocks

With DASL only:

For installation: 48,000 blocks

For use (permanent): 35,500 blocks

With M WAPI and Windows only:

For installation: 28,000 blocks

For use (permanent): 23,500 blocks

With DASL, M WAPI and Windows:

For installation: 50,000 blocks

For use (permanent): 37,500 blocks

These counts refer to the disk space required on the system disk. The sizes are approximate; actual sizes may vary depending on the user's system environment, configuration, and software options.

Cluster Environment

DSM for OpenVMS is fully supported when installed on any valid and licensed VMScluster¹ configuration, without restrictions. The **Hardware Requirements** sections of this product's Software Product Information detail any special hardware required by this product.

¹ VMScluster configurations are fully described in the VMScluster Software Product Description (29.78.xx) and include CI, Ethernet, and Mixed Interconnect configurations.

DSM for OpenVMS Version 6.5

Software Requirements

For Alpha Systems:

OpenVMS Alpha Operating System V6.1 or greater

For VAX Systems:

OpenVMS VAX Operating System V6.1 or greater

OpenVMS Tailoring

The following OpenVMS classes are required for full functionality of this layered product:

- OpenVMS Required Saveset
- System Programming Support
- Miscellaneous Files
- Utilities
- Network Support (if DECnet DDP capability is required)
- Programming Support
- DECwindows Motif (if DECwindows Motif support is required)

Optional Software

For Alpha Systems:

- DECwindows Motif for OpenVMS Alpha V1.1
- DEC TCP/IP Services for OpenVMS V3.0 or greater
- WIN/TCP V5.2™
- TCPware for OpenVMS V5.0 or greater
- DECimage Application Services for VMS²

For VAX Systems:

- VMS DECwindows Motif V1.0
- DEC TCP/IP Services for OpenVMS VAX V2.0 or greater
- WIN/TCP V5.2™
- TCPware™ for OpenVMS V5.0 or greater
- DECimage Application Services for VMS²

Growth Considerations

The minimum hardware/software requirements for any future version of this product may be different from the requirements for the current version.

² Any version compatible with the version of DECwindows used.

Distribution Media

DSM for OpenVMS is distributed by both InterSystems Corporation and Digital Equipment Corporation.

DSM for OpenVMS VAX and DSM for OpenVMS Alpha are available from InterSystems Corporation on 9-track 1600 BPI Magtape, TLZ06 cartridge, and TK50 Streaming Tape.

DSM for OpenVMS VAX is available from Digital Equipment Corporation on 9-track 1600 BPI Magtape or TK50 Streaming Tape. It is also available as part of the OpenVMS Consolidated Software Distribution on CD-ROM.

DSM for OpenVMS Alpha is available from Digital Equipment Corporation only on the Digital CD-ROM Software Library for OpenVMS Alpha.

Ordering Information

Please contact InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142 at 617-261-0600 for more information.

Order Processing

For questions about orders, contact InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142 at 617-577-3600.

Software Licensing

For more information about the licensing terms and policies, contact InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142 at 617-261-0600.

License Management Facility

This layered product supports the OpenVMS License Management Facility (LMF). License units for this product are allocated on a Concurrent Use basis. Each Concurrent Use license allows any one DSM process at a time to use the layered product.

For more information about the OpenVMS License Management Facility, refer to the appropriate OpenVMS Operating System Software Product Description or documentation or contact InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142 at 617-621-0600.

Software Warranty

Warranty for this software product is provided by InterSystems with the purchase of a license for the product.

DSM for OpenVMS Version 6.5

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

TCPware for OpenVMS is a trademark of Process Software Corporation.

DSM for OpenVMS VAX is a trademark of InterSystems Corp.

DSM for OpenVMS Alpha is a trademark of InterSystems Corp.

WIN/TCP is a trademark of The Wollongong Group.

Alpha, DASL, DEC, DECimage, DECnet, DECstation, DECsystem, DECwindows, OpenVMS, Digital UNIX, VAX, VMScluster, VAXstation, OpenVMS, VMS RMS, VMScluster, DEC TCP/IP for OpenVMS (UCX) and the DIGITAL logo are trademarks of Digital Equipment Corporation.

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

© InterSystems Corporation, 1996. All rights reserved.