# Software Product Description

**PRODUCT NAME: VAXELN Toolkit, Version 4.2** **SPD 28.02.12**

## DESCRIPTION

The VAXELN Toolkit is a VMS layered product used for the development of dedicated realtime VAXELN systems that run on VAX and MicroVAX processors. The development tools run on any host VAX computer under VMS or MicroVMS Operating Systems. A finished VAXELN system runs directly on a supported runtime target VAX or MicroVAX processor without the presence of another operating system.

VAXELN is especially suited to, although not limited to, creating realtime applications; that is, applications in which the system's response (both speed and predictability) to external events is critical.

The VAXELN software simplifies the design and implementation of such applications by offering high-level language support, a runtime kernel executive and various service programs. The high level language support includes a Pascal compiler and runtime libraries (RTL) for VAX C and VAX FORTRAN. Support for VAX Ada is available under separate license. The kernel manages resources, processes and shared data. The runtime service programs available are for a VMS compatible file system, network communication facilities and device drivers.

VAXELN supports multi-tasking; that is, execution of a program made up of several concurrently executing parts. In addition, multiprogramming is supported, meaning that entire programs, including multi-tasking programs, can be scheduled concurrently on the same CPU.

VAXELN provides support for tightly coupled symmetric multi-processing on the VAX 6000 Model Series 200, 300 and 400, and VAX 8800 configurations. Each processor executes a single copy of the system image and jobs can run on either processor.

VAXELN DECwindows is based on M.I.T.'s specification for the X Window™ System, Version 11, Release 3. X Window System standards supported as part of DECwindows include the X11 network protocol, a base set of workstation fonts, the C language binding for the Xlib programming library and the C language binding for the Xtoolkit library. Support of the X11 network protocol in the client, library and display server components provides VAXELN with the capability to interoperate with other X11-compliant systems in a distributed fashion. VAXELN supports the server-client distribution inherent in the X Window System with VAXELN provided transports - local shared memory, DECnet or TCP/IP.

VAXELN DECwindows provides full client/server functionality. Provided are the DECwindows server image, the DECwindows runtime libraries and the user environment component DECwindows software, all of which can be built into VAXELN systems.

*VAXELN Systems*

VAXELN systems can run on stand-alone VAX and MicroVAX computers, or with networking software provided in the Toolkit; they can be connected in an Ethernet local area network (LAN). The latter may include VMS nodes or any other nodes using the Digital Network Architecture DECnet services and protocols.

VAXELN systems, including device drivers, can be developed entirely in a high-level language. The sources for several Digital implemented device drivers are also included. These can be used as a guide to implementing additional drivers.

A VAXELN system can be booted from a hard disk, tape, diskette or, if the VMS host system has the optional DECnet-VAX license and Ethernet hardware, by downline loading the system into the target computer. VAXELN system images are also suitable for placement in erasable programmable read-only memories (EPROMs). However, the EPROM hardware and software are not part of the Toolkit and must be acquired separately.

*Toolkit Components*

- VAXELN Pascal Compiler
- VAXELN Symbolic Debugger
- VAXELN System Builder
- VAXELN Kernels for Supported Target VAX CPUs
- VAXELN Pascal Runtime Library

- VAXELN C Runtime Library
- VAXELN FORTRAN Runtime Library
- VAXELN Device Drivers for Supported Devices
- VAXELN File Service
- VAXELN Network Service
- VAXELN Authorization Service
- VAXELN Performance Analyzer
- VAXELN Resource Monitor and Display
- VAXELN Command Language
- VAXELN Error Logging Utilities
- VAXELN Local Area Terminal (LAT) Service
- VAXELN DECwindows Server
- VAXELN DECwindows User Environment Components
- VAXELN User Documentation
- VAXELN TCP/IP Networking
- VAXELN Set Host

A VAXELN application image executes in its own runtime environment. Programs written to take advantage of VMS or ULTRIX system-specific features (i.e., logical names, QIO, RMS) need to be modified in order to be ported to the VAXELN Runtime environment.

The VAXELN Pascal language is based upon the ANSI/IEEE770X3.97-1983, American National Standard Pascal Computer Programming Language. It is supported by an optimizing compiler that generates position-independent, native-mode code. The Pascal language was extended to allow its use for VAXELN system programming, and to eliminate the need to use assembly language for VAXELN system programming.

The VAXELN C Runtime Library implements the standard features of the C language plus VAXELN-specific capabilities designed to assist the C program developer in taking full advantage of VAXELN. Routines selected for inclusion in this library are primarily those which an efficient emulation of standard UNIX® C is possible under VAXELN. VAXELN C uses the standard VMS C compiler.

The VAXELN FORTRAN Runtime Library supports an extensive subset of the features of the standard VAX FORTRAN language. This library supports I/O to sequential devices and sequential and direct access to sequential files on mass storage media. The routines in this library also allow the FORTRAN programmer to access kernel services for developing realtime, multiprogramming applications. Since FORTRAN does not generate re-entrant code, care must be exercised to avoid multiple concurrent threads executing through subroutines and/or functions. That is, VAX FORTRAN subroutines and/or functions are serially reusable, but not re-entrant.

User code is linked to VAXELN Runtime Libraries using the standard VMS linker. The libraries are provided in both object-library and shareable-image form.

The VAXELN System Builder combines program images, the VAXELN kernel image, and runtime routines to produce the finished VAXELN system. The program loader permits dynamic loading of individual program images.

The VAXELN Debugger can be used to debug a VAXELN system locally using the target system's console terminal. If the user's host system has the optional DECnet-VAX license, it can be used remotely to debug VAXELN systems running on the same DECnet Ethernet network.

The VAXELN Kernel manages the system's processes and data providing the controlled sharing of the system's resources. VAXELN systems are memory resident utilizing VAX virtual memory hardware. The VAX page fault mechanism is used only for dynamic extension of the user stack. A VAXELN system is limited in size by the amount of physical memory present on the target system.

The VAXELN File Service is a set of services to enable using disk or tape files for program I/O. The File Service consists of a Disk File Service and a separate Tape File Service. The Disk File Service uses the same Files-11 on-disk structure, ODS-2, as does VMS and the same internal data format as the VAX Record Management Services (RMS). Consequently, disk volumes can be read or written from one environment to another. File Service files are sequentially organized, but can be accessed either sequentially or randomly. The indexed and relative file organizations are not supported by VAXELN. VAXELN systems and VMS systems running on nodes in the same network can access files on each other's disk volumes. The Tape File Service is based on Version 3 of the ANSI standard for magnetic tapes. It is compatible with the VMS, Version 4.0 tape file system, providing users with a convenient means of transporting files to and from VMS systems.

The VAXELN Network Service is responsible for providing transparent network message transmissions between CSMA/CD LAN (Ethernet/IEEE802.3) nodes. The Network Service uses the Phase IV DECnet Routing Protocol, Version 2.0, to route system-level datagrams between VAXELN nodes and other DECnet nodes. (Only end-node functionality is provided). The VAXELN system can communicate over the CSMA/CD

LAN with any DECnet node on the same LAN. If there is a full routing system on the LAN, VAXELN can communicate with any other DECnet node in the entire network. The VAXELN CSMA/CD LAN driver will support multiple controllers of the same type simultaneously. At any given time, DECnet can run on only one controller. Other CSMA/CD LAN protocols can be implemented on the other controllers. VAXELN supports the dynamic enabling and disabling of DECnet. In the event of hardware failure, this feature can be used to dynamically disable DECnet on one controller, followed by enabling DECnet on another controller. Downline loading over the CSMA/CD LAN of a VAXELN system image, from a VAXELN load host to a VAXELN target, is supported.

The Phase IV DECnet Network Services Protocol (NSP), Version 4.0 and the Session Control Protocol, Version 1.0 are used to provide transparent application-level circuits with remote nodes. The Network Service's NSP module then uses the Routing Protocol to deliver messages to remote systems.

VAXELN DECnet nodes can be managed from VMS with the DECnet-VAX Network Control Program (NCP). The Network Service supports the Phase IV Network Management Protocol, Version 4.0 Subset.

VAXELN supports industry standard networking through a set of ARPANET protocols. The supported protocols include: Transmission Control Protocol (TCP), Internet Protocol (IP), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), User Datagram Protocol (UDP) and Boot Protocol (BOOTP). This support is provided over a CSMA/CD LAN (Ethernet) and it interoperates with ULTRIX, VMS/ULTRIX Connection and UNIX. C Language applications and may be written using the Berkeley Socket Programming interface.

The optional VAXELN Authorization Service maintains a database of all the users authorized to use a particular VAXELN system or network of systems and is compatible with standard, FILES-11 protection and security features.

In conjunction with the VMS Error Log Utility, optional error logging is supported. The error log file can be created locally on the target system or created remotely on a VMS node in the network. The error log file is analyzed on a VMS system using the VMS Error Reporting Facility.

VAXELN Command Language (ECL) provides commands to communicate with the VAXELN system. VAXELN ECL provides file manipulation, program manipulation and limited system control functions on the VAXELN system.

VAXELN Resource Monitor (EDisplay) dynamically displays information about the VAXELN system. This information includes pool and memory usage, and currently active jobs and processes. Help feature functionality is included.

VAXELN Performance Analyzer (EPA) provides collection of information and analysis of the runtime behavior of the VAXELN system. The information is subsequently analyzed on a VMS host.

Local area transport (LAT) host services allow VAXELN targets to communicate with devices attached to terminal servers. These services include terminal server communication, terminal I/O, and a control interface for application programs plus an interactive utility, which can be used to control and monitor a LAT environment.

The VAXELN DECwindows Server provides a common means for DECwindows applications to interact with graphics workstations. The DECwindows server image runs as a job on the hardware where the display, keyboard and mouse are located in a VAXELN runtime system.

The VAXELN DECwindows Client provides runtime routines and tools for the development of applications using capabilities of graphics workstations.

The VAXELN DECwindows User Environment Components provides Window Manager and terminal emulation support.

*VAXELN Kernel Objects and Operations*

Processes are the execution threads for VAXELN programs. The main thread of execution (master process) corresponds to the program main routine.

Programs can also include special process blocks that are executed by dynamically created subprocesses. The family of a master process and all of its subprocesses is a job in VAXELN terminology. One copy of a program's code is shared by any number of jobs executing that program. Jobs and processes are scheduled on a pre-emptive priority basis.

Ports are first-in/first-out queues of messages that are exchanged between processes in a VAXELN application. Port names are provided to ensure the transparency of message-port locations in the local area network. Circuits are bidirectional message paths between two specified ports. Circuits guarantee delivery, blocking and proper sequencing of network messages, and flow control (prevention of a slower receiver process being overwhelmed by a faster sender).

Where memory sharing between jobs is a necessity and where the network distribution of the jobs is not an issue, VAXELN provides the AREA object as a means of sharing data among jobs on a single VAXELN node. Events and semaphores are synchronization objects used by multiple processes within a job. Devices are synchronization objects used to synchronize a device driver program with the completion of an interrupt service routine.

**SOURCE CODE**

Optional source materials for this software product can be obtained by signing Digital's Software Program Sources License Agreement and then purchasing the desired source option(s). The agreement entitles the user to use the source materials at one customer facility or location which is specified in the agreement.

Included in the source kit are the kernel and necessary device drivers for the rtVAX 300 processor. If the rtVAX 300 is to be embedded into a board with an I/O bus, or used in an application that will utilize DMA operations, customization of the VAXELN kernel to support the full environment may be required. Source components and documentation for kernel customization are included with the VAXELN source kit.

Most users do not require source materials which are used primarily to make modifications to the software product. Source kits provided by Digital do not contain all source files used by Digital to build binary kits.

**HARDWARE REQUIREMENTS**

VAX, MicroVAX or VAXstation configurations as specified in the System Support Addendum (SSA 28.02.12-x).

**SOFTWARE REQUIREMENTS***

*For Host Development System*

- VMS Operating System

  or

- MicroVMS Operating System

* Refer to the System Support Addendum (SSA 28.02.12-x) for availability and required versions of prerequisite/optional software.

**ORDERING INFORMATION**

*For VAXELN Toolkit*

Software Licenses: QL-375A*-**
Software Media: QA-375A*-**
Software Documentation: QA-375AA-GZ
Software Product Services: QT-375A*-**

*For VAXELN Runtime System*

Software Licenses: QL-376A*-**

* Denotes variant fields. For additional information on available licenses, services and media, refer to the appropriate price book.

**SOFTWARE LICENSING**

The Single-Use License is required for each CPU that will be using the VAXELN Toolkit to support development of VAXELN systems.

If the VAXELN system is developed and used on the same CPU that is a valid VAXELN target system, then a separate Single-Use VAXELN Target System License is not needed. However, if the application (VAXELN System) is developed on one CPU and then loaded onto one or more additional CPUs, a Single-Use VAXELN Target System License is needed for each additional target system CPU.

Single-Use License Option For Host Development Systems (QL-375A*-A#):

This Single-Use License is your right to use the VAXELN Toolkit on a single CPU.

*For VAXELN Target Systems (QL-376A*-FA):*

*Single-Use License Option*

If the VAXELN Toolkit is installed on a host CPU fully licensed for that product and you wish to use the VAXELN Runtime System portion of it on an additional CPU, a Single-Use VAXELN Target License can be purchased. This license provides the right to use only the VAXELN Runtime System on that target system and does not include a software warranty.

*Source License and Sources Distribution Option*

This option supplies the machine-readable source code for this software product. It gives you the right to use the source code on any CPU at the facility/location specified in the agreement which has a Single-Use License for the object code.

The Source Kit does not contain all sources needed to build the product.

Specifically excluded are sources of a proprietary nature (e.g., T/MSCP and DAP) and sources for any of the compilers.

This software is furnished under the licensing provisions of Digital Equipment Corporation's Standard Terms and Conditions. For more information about Digital's licensing terms and policies, contact your local Digital office.

## LICENSE MANAGEMENT FACILITY SUPPORT

This layered product supports the VMS License Management Facility.

License units for this product are allocated on a CPU-capacity basis.

For more information on the License Management Facility, refer to the VMS Operating System Software Product Description (SPD 25.01.xx) or the License Management Facility of the VMS Operating System documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

## SOFTWARE PRODUCT SERVICES

A variety of service options are available. For more information, please contact your local Digital office.

## SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.

® UNIX is a registered trademark of American Telephone and Telegraph Company.

™ X Window is a trademark of Massachusetts Institute of Technology.

™ The DIGITAL Logo, VAXELN, VAX, VMS, MicroVAX, VAXstation, VAXserver, VAXcluster, DECwindows and DECnet-VAX are trademarks of Digital Equipment Corporation.