Software Product Description

PRODUCT NAME: DECmessageQ for OpenVMS, Version 3.2 **SPD 46.25.04**

DESCRIPTION

DECmessageQ for OpenVMS is the OpenVMS (VAX and Alpha) operating system implementation of a generic software message queuing bus that provides easy-to-use, efficient task-to-task communications among processes using DECmessageQ on OpenVMS, ULTRIX, MS–DOS/2®, Macintosh®, HP®-UX, AIX®, 88open®, and Digital UNIX®, Windows NT™ (Alpha and Intel®), and SunOS® and Solaris®.

A common call interface allows messages to be delivered via high-speed global sections for some intra-CPU applications, or via DECnet-VMS and Transmission Control Protocol/INTERNET Protocol (TCP/IP) for inter-CPU applications. Applications can be designed so client applications can be redeployed easily anywhere within the DECmessageQ network configuration, whether in a standalone node, a Local Area Network (LAN), or a Wide Area Network (WAN).

The DECmessageQ communication implementation is designed for ease-of-use, expandability, and efficiency.

Features

- High speed local message delivery using Global Sections
- Remote message delivery via DECnet for MS–DOS, Macintosh, OS/2, Digital UNIX, ULTRIX, Windows NT and OpenVMS connections; and via TCP/IP for UNIX/ULTRIX, Windows NT and OpenVMS connections.
- Remote message delivery to MS-Windows via DECnet or TCP/IP using Client Library Services (CLS)
- Fully asynchronous and synchronous receipt of messages
- Message sizes up to 32,000 bytes
- Priority queuing of messages
- Message recovery using Message Recovery Services (MRS)
- Message broadcasting using Selective Broadcast Services (SBS)

- PC connections to Client Library Services (CLS)
- Selective reception of messages according to certain fields within the DECmessageQ header and within the data portion of the message
- Shared input queues using Multi-Reader Queues (MRQ)
- Use of intra-CPU naming through high-speed global sections, and inter-CPU naming through DECdns distributed name services
- Message simulation facility for use in application testing
- Set of message delivery options
- Selectable tracing of messages and calls to DECmessageQ
- A maximum of 999 queues per DECmessageQ Group
- A maximum of 32,000 DECmessageQ Groups
- Integration of OpenVMS timers
- Support for languages adhering to the OpenVMS Calling standard
- Utilities for monitoring the network configuration and flow of messages
- Dynamic addition of message queues and CPUs to the communication network
- Message interface for retrieving DECmessageQ configuration information on-line
- Portable call interface
- Connectivity to DECmessageQ implementations on OpenVMS, ULTRIX, Windows NT, OS/2, MS–DOS, MS-Windows, Macintosh, HP-UX, AIX, Digital UNIX, 88open, and SunOS and Solaris.

DECmessageQ for OpenVMS can also be used as an integration tool to merge many external events with the Message Queuing Bus. In addition to integrating messages from local processes and remote processes, DECmessageQ also provides facilities for the integration of other events such as:

- Timer settings
- Simulated messages
- LU6.2 conversations
- Other external events such as special hardware I/O

DECmessageQ Queues

DECmessageQ for OpenVMS provides three types of queues. A queue is a memory storage location for DECmessageQ messages. Any process can insert a message into any queue. These queues are accessed directly by DECmessageQ procedures. These procedures are called by user-written applications. DECmessageQ queue types are:

- Primary Queue (PQ) — Each process that attaches to the message queuing bus is assigned a Primary Queue. This queue is used to receive messages from processes using DECmessageQ.
- Secondary Queue (SQ) — Any process may attach to one or more secondary queues. These queues can also be used to receive messages. The order in which queues are scanned for messages is defined by the DECmessageQ selection rules.
- Multi-Reader Queue (MRQ) — A Multi-Reader Queue is a single shared queue that can be shared by multiple processes.

Message Recovery Services (MRS)

Message Recovery Services for the DECmessageQ Message Queuing Bus extend data recovery to the level of pending messages. Using Message Recovery Services, the sender is relieved of the responsibility of tracking the progress of a message through its next level of processing. This functionality can be used both at the client and server sides of the application.

Message Recovery Services increase the robustness of DECmessageQ by providing applications with the ability to recover from message delivery failures due to:

- Application program abort
- Communication line failure
- System crash

Some of the applications requirements addressed by MRS are:

- Sender wishes to insure delivery of messages when the receiving process is available but does not wish to monitor the delivery.
- Sender wishes to know that a message is recoverable to avoid the cost of reconstructing it but does not care when it is finally delivered.
- Receiver wishes to maintain a journal of all messages received by it for audit trail or reprocessing.

Message Recovery Services are primarily implemented by an MRS server, a non-privileged program attached to the DECmessageQ Message Bus. MRS actions are invoked by standard DECmessageQ send and receive message calls.

MRS is oriented toward messages, not processes; not all messages sent from or directed to a particular process need to be processed by MRS. This allows applications to selectively incur the additional processing imposed by MRS for just those messages that are not easily recovered. Message recovery characteristics are set by the sending process.

Exchange of recoverable messages with non-OpenVMS systems requires a minimum of DECmessageQ V3.0A.

Selective Broadcast Services (SBS)

There are two important application requirements handled by this service. The first requirement is the ability to send a message to many targets without going through multiple application DECmessageQ send message requests. The second requirement is the ability to generate broadcast messages without the originator knowing the quantity or location of the target process(es).

Selective Broadcast Services provide a conceptual broadcast stream of data. In this broadcast stream any process can insert a message. Any process can select messages from this broadcast stream for delivery. Messages may be selected using a set of rules that provide relational comparisons against DECmessageQ header information or user's message data information. The Selective Broadcast Services operate in a single server environment or between cooperating DECmessageQ servers. When the SBS is operating between nodes, it can operate using DECnet or TCP/IP services, or using direct Ethernet operations. In the direct Ethernet operations, the SBS will optimize message traffic by using Ethernet's multi-casting capabilities.

Delivery Options

DECmessageQ for OpenVMS provides the following set of delivery options:

- Datagram — A non-recoverable attempt is made to deliver a message. If the message cannot be delivered to a target, then an error is logged.
- Blocking and non-blocking enqueue — The sending process will be notified when the message is written to the target queue. A return status will indicate if the message successfully enqueued to the queue.
- Blocking and non-blocking dequeue - The sending process will be notified when the message is read from the target queue. A return status will indicate if the message successfully dequeued from the queue.
- Blocking and non-blocking Acknowledge - The sending process will be notified when the target process confirms the message. A return status will indicate if the message was successfully confirmed by the target.
- Blocking and non-blocking Recoverable - The sending process will be notified when the recovery system has accepted the message. A return status will indicate if the message was successfully journalled by the recovery system.
- Blocking and non-blocking Recoverable Acknowledge - The sending process will be notified when the target process has confirmed the recoverable message. A return status will indicate if the message was successfully confirmed by the target process.

DECmessageQ Scripts Facility

The DECmessageQ scripts facility provides a general capability to perform message capture, simulation, and replay.

- Capture — DECmessageQ scripts can be used to capture messages sent or received from a process. These messages can be displayed on the output device or collected in a disk file. The messages are displayed in an ASCII file using the DECmessageQ scripts syntax.
- Simulation — A disk file containing DECmessageQ scripts commands can be used to simulate message traffic to a process. The commands use the DECmessageQ scripts syntax. The process will receive these messages after any DECmessageQ queue is scanned for receivable messages.
- Replay — A replay is the simulation of messages previously captured.

DECmessageQ LU6.2 Services (License Option)

DECmessageQ LU6.2 Services for OpenVMS systems allows user programs attached to a DECmessageQ Message Queuing Bus to request, accept, and conduct APPC (Advanced Program-to-Program Communications) conversations with programs running under CICS/VS, IMS/VS, and other IBM® environments over SNA LU6.2 sessions.

Note

Refer to the DECnet/SNA APPC/LU6.2 Programming Interface for OpenVMS Software Product Description (SPD 26.88.xx) for currently supported environments.

DECmessageQ LU6.2 Services allows users unfamiliar with SNA communications to develop, quickly and easily, applications that operate with IBM-based APPC applications. Users are insulated from the details of both SNA and APPC. DECmessageQ provides additional insulation between the two environments by translating transaction program names known to DECmessageQ user programs to names known to the IBM system(s) and vice versa.

DECmessageQ LU6.2 Services for OpenVMS consists of:

- An LU6.2 Server image ("Port Server") that provides connection management facilities, isolating the end user from details of SNA-APPC communications
- A library of predefined DECmessageQ message structures for ease of program development
- Utilities for managing the LU6.2 Server
- Tools for developing user-written LU6.2 Servers

DECmessageQ LU6.2 Services Limits

DECmessageQ LU6.2 Services for OpenVMS has the following limits:

Maximum user message size:	31,982 bytes
Maximum number of active LUs:	256 per Port Server
Maximum number of remote Transaction Programs:	512 per DECmessageQ LU6.2 Server (user-written servers: unlimited)

DECmessageQ LU6.2 Services Utilities

DECmessageQ LU6.2 Services for OpenVMS includes a management utility that allows a suitably privileged user to start, stop, and configure DECmessageQ LU6.2 servers and observe their activity (through an event monitoring facility).

Users can develop specialized LU6.2 servers by using the DECmessageQ LU6.2 Server tool kit. The tool kit consists of a DECmessageQ for OpenVMS User Callback (a "user callback" is a special User Exit module) and 21 predefined DECmessageQ message structures.

The LU6.2 Server tool kit gives the user a finer degree of control over the APPC environment than the standard DECmessageQ LU6.2 Server. Functions supported via the tool kit are:

- DEFINE_LU
- DEFINE_TP
- DELETE_LU
- ALLOCATE_CONVERSATION
- DEALLOCATE_CONVERSATION
- REQUEST_CONFIRMATION
- ISSUE_CONFIRMATION
- ISSUE_ERROR
- REQUEST_TO_SEND
- SEND
- RECEIVE

HARDWARE REQUIREMENTS

EISA Bus Based Systems

- DEC 2000 Models 300/500

TurboChannel Bus Based Systems

- DEC 3000 Models 300/300L/300LX
- DEC 3000 Models 400/400S
- DEC 3000 Models 500/500S/500X
- DEC 3000 Models 600/600S
- DEC 3000 Models 700/700LX
- DEC 3000 Models 800/800S
- DEC 3000 Models 900/900LX

DSSI Bus Based Systems

- DEC 4000 Model 600
- DEC 4000 Model 700

XMI Bus Based Systems

- DEC 7000 Model 600
- DEC 10000 Model 600

PCI Bus Based Systems

- Digital 2100 Server Model A500MP
- Digital 2100 Server Model A600MP
- AlphaServer 2100 4/233
- AlphaServer 2000 4/233
- AlphaServer 1000 4/200
- AlphaServer 1000 4/166
- AlphaServer 8200 5/300
- AlphaServer 8400 5/300
- AlphaServer 2100 5/250
- AlphaStation 400 4/266
- AlphaStation 250 4/233

- AlphaStation 200 4/233
- AlphaStation 200 4/166
- AlphaStation 600 5/266

Q–bus Based Systems

- MicroVAX II
- VAXstation II/GPX[1], VAXstation II/QVSS[2]
- MicroVAX 3200, VAXserver 3200, VAXstation 3200
- MicroVAX 3300, VAXserver 3300
- MicroVAX 3400, VAXserver 3400
- MicroVAX 3500, VAXserver 3500, VAXstation 3500, VAXstation 3520, VAXstation 3540
- MicroVAX 3600, VAXserver 3600
- MicroVAX 3800, VAXserver 3800
- MicroVAX 3900, VAXserver 3900
- VAX 4000, Models 100, 200, 300, 500, 600
- VAX 4000, Models 50, 100A, 500A, 600A, 700A
- VAXserver 4000, Models 200, 300, 400, 500, 600

NMI Bus Based Systems

- VAX 8530, VAXserver 8530, VAX 8550, VAXserver 8550
- VAX 8700, VAXserver 8700
- VAX 8800, VAX 8810, VAX 8820, VAX 8830, VAX 8840
- VAXserver 8800, VAXserver 8810, VAXserver 8820, VAXserver 8830, VAXserver 8840

XMI Bus Based Systems

- VAX 6000 Series, Models 210, 220, 230, 240
- VAX 6000 Series, Models 310, 320, 330, 340, 360
- VAX 6000 Series, Models 410, 420, 430, 440, 450, 460
- VAX 6000 Series, Models 510, 520, 530, 540, 550, 560

[1] Graphics Processing Accelerator (GPX)

[2] Q–bus Video Sub System (QVSS)

- VAX 6000 Series, Models 610, 620, 630, 640
- VAXserver 6000, Models 210, 220, 310, 320, 410, 420, 510, 520
- VAX 8530, VAX 8550, VAXserver 8530, VAXserver 8550
- VAX 8700, VAXserver 8700
- VAX 8800, VAX 8810, VAX 8820, VAX 8830, VAX 8840
- VAXserver 8800, VAXserver 8810, VAXserver 8820, VAXserver 8830, VAXserver 8840
- VAX 7000, Models 610, 620, 630, 640, 650, 660
- VAX 10000, Models 610, 620, 630, 640, 650, 660

VAXBI Bus Based Systems

- VAX 8200, VAX 8250, VAXserver 8200, VAXserver 8250
- VAX 8300, VAX 8350, VAXserver 8300, VAXserver 8350

SBI Bus Based Systems

- VAX 11/780, VAX 11/785
- VAX 8600, VAX 8650, VAXserver 8600, VAXserver 8650

CMI Bus Based Systems

- VAX 11/750

Special System Specific Internal Bus

- VAX 11/730
- MicroVAX 2000, VAXstation 2000, VAXstation 2000/GPX, VAXstation 2000/MFB[3]
- VAXft 3000, Models 110, 310, 410, 610, 612
- MicroVAX 3100, Models 10, 10E, 20, 20E, 30, 40, 80, 85, 90, 95
- VAXserver 3100, Models 10, 10E, 20, 20E
- VAXstation 3100, Models 30, 38, 40, 48, 76

[3] Monochrome Frame Buffer (MFB)

- VAXstation 3100/GPX, Models 38, 48, 76
- VAXstation 3100/SPX[4], Models 38, 48, 76
- VAXstation 4000, Models 60, 90
- VAXstation 4000-VLC
- VAX 9000, Models 110, 110VP[5], 210, 210VP, 310, 310VP
- VAX 9000, Models 320, 320VP, 330, 330VP, 340, 340VP
- VAX 9000, Models 410, 410VP, 420, 420VP, 430, 430VP
- VAX 9000, Models 440, 440VP

DECmessageQ is not supported on these systems

MicroVAX I/VAXstation I Systems
VAX–11/725
VAXstation 8000

Processor Restrictions

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

Disk Space

The following table summarizes DECmessageQ disk space requirements:

[4] 2D Scanline Processor Accelerator Graphics System (SPX)
[5] Vector Processor (VP)

Kit	Blocks Used During Installation	Blocks Required After Installation
Alpha Platform		
OpenVMS Alpha Development	115,000	110,000
OpenVMS Alpha Run-Time only	45,000	40,000
OpenVMS Help Library (optional)		1,000
TCP/IP support (optional)		1,000
LU6.2 support (optional)		10,000
VAX Platform		
OpenVMS VAX Development	50,000	45,000
OpenVMS VAX Run-Time only	35,000	20,000
OpenVMS Help Library (optional)		1,000
TCP/IP support (optional)		1,000
LU6.2 support (optional)		10,000

These counts refer to the disk space required on the system disk, or the installation target disk if other than the system disk, for the permanent storage requirement. The sizes are approximate; actual sizes may vary depending on the user's system environment, configuration, and software options.

OPTIONAL HARDWARE

- Any communication device capable of supporting DECnet, DEC TCP/IP Services for OpenVMS, TGV Multinet® or Wollongong PathWay™.
- Any Ethernet controller supported by the OpenVMS operating system.

CLUSTER ENVIRONMENT

This layered product is fully supported when installed on any valid and licensed VMScluster* configuration without restrictions. The *HARDWARE REQUIREMENTS* section details any special hardware required by this product.

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

SOFTWARE REQUIREMENTS

For Systems Using Terminals (No DECwindows Interface):

- OpenVMS Alpha Operating System V1.5 - V7.0
- OpenVMS VAX Operating System V5.5-2 - V7.0
- DECnet/SNA APPC/LU6.2 Programming Interface for OpenVMS V2.2, at a minimum (for LU6.2 Services option only)

For OpenVMS VAX systems, the following OpenVMS classes are required for full functionality of this layered product:

- OpenVMS Required Saveset
- Network Support
- Programming Support
- System Programming Support
- Secure User’s Environment

For more information on OpenVMS classes and tailoring, refer to the OpenVMS VAX Operating System Software Product Description (SPD 25.01.xx).

OPTIONAL SOFTWARE

DEC TCP/IP Services for OpenVMS VAX V3.1 or higher
or
TGV MultiNet V3.3A or higher
or
Wollongong Pathway V1.1.1.1

- DECnet-Alpha V1.5 and higher
- DECnet–VAX V5.5 and higher
- DECnet/OSI V5.8 and higher
- VAX Distributed Name Service
- DECdns client on OpenVMS Alpha systems which requires DECnet/OSI
- DECmessageQ for ULTRIX Client V1.0
- DECmessageQ for ULTRIX V3.0A*
- DECmessageQ for MS–DOS® Client V2.0
- DECmessageQ for MS-Windows® Client V3.0

- DECmessageQ for OS/2 Client V2.0
- DECmessageQ for Macintosh Client V2.0
- DECmessageQ for AIX V3.0A*
- DECmessageQ for HP-UX V3.0A*
- DECmessageQ for Digital UNIX V3.0A*
- DECmessageQ for 88open V3.0A*
- DECmessageQ for SunOS V3.0A*
- DECmessageQ for Sun Solaris V3.0A*

* Information about these products can be found in the DECmessageQ for UNIX® Software Product Description (SPD 39.25.xx).

GROWTH CONSIDERATIONS

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

DISTRIBUTION MEDIA

9-track 1600 BPI Magtape, TK50 Streaming Tape (OpenVMS VAX only), CD–ROM (OPENVMS Alpha only).

This product is also available as part of the OpenVMS (VAX and Alpha) Consolidated Software Distribution on CD–ROM.

The software documentation for this product is also available as part of the OpenVMS (VAX and Alpha) Online Documentation Library on CD–ROM.

ORDERING INFORMATION

Development Option

The DECmessageQ for OpenVMS Development Option includes the base message queuing software, language compiler binding libraries, and configuration definition utilities.

Run-Time Only Option

The Run-Time Only Option provides all the facilities of the development version of DECmessageQ for OpenVMS except the language compiler binding libraries and the configuration definition utilities. The purpose of the Run-Time Only version is to support the execution of previously developed applications on a target machine. Program development is not supported under the Run-Time Only Option.

LU6.2 Services Option

The DECmessageQ for OpenVMS LU6.2 Services option supports conversations with programs running under CICS/VS, IMS/VS, and other IBM environments over SNA LU6.2 sessions.

Software Licenses

DECmessageQ for OpenVMS VAX Development Only:
QL-GKPA9-J*
DECmessageQ for OpenVMS VAX Run-Time Only:
QL-GHVA9-J*
DECmessageQ LU6.2 Services for OpenVMS VAX:
QL-GKQA9-J*
DECmessageQ for OpenVMS Alpha Development Only:
QL-0HJA*-AA
DECmessageQ for OpenVMS Alpha Run-Time Only:
QL-0HKA*-AA
DECmessageQ LU6.2 Services for OpenVMS Alpha:
QL-0HLA*-AA

Software Media/Documentation

DECmessageQ for OpenVMS VAX and
DECmessageQ LU6.2 Services for OpenVMS VAX:
QA-GKPAA-H*
DECmessageQ for OpenVMS Alpha and
DECmessageQ LU6.2 Services for OpenVMS Alpha:
QA-0HJAA-H*

Software Documentation

All DECmessageQ OpenVMS VAX Products:
QA-GKPAA-GZ
All DECmessageQ OpenVMS Alpha Products:
QA-0HJAA-GZ

Software Product Services

DECmessageQ for OpenVMS VAX Development:
QT-GKPA*-**
DECmessageQ for OpenVMS VAX Run-Time Only:
QT-GHVA*-**
DECmessageQ LU6.2 Services for OpenVMS VAX:
QT-GKQA*-**
All DECmessageQ for OpenVMS Alpha Development Only:
QT-0HKA*-*

DECmessageQ for OpenVMS Alpha Run-Time Only:
QT-0HKA*-**
DECmessageQ LU6.2 Services for OpenVMS Alpha:
QT-0HLA*-**

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

The above information is valid at time of release. Please contact your local Digital office for the most up-to-date information.

® 88open is a registered trademark of 88open Consortium, Ltd.

® IBM, AIX, and OS/2 are registered trademarks of International Business Machines Corporation.

® HP is a registered trademark of Hewlett-Packard Corporation.

® Macintosh is a registered trademark of Apple Computer, Inc.

® MS and MS–DOS are registered trademarks of Microsoft Corporation.

® MultiNet is a registered trademark of TGV, Inc.

® Sun, SunOS, and Solaris are registered trademarks of Sun Microsystem Corporation.

® UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

™ Windows and Windows NT are trademarks of Microsoft Corporation.

™ Intel is a trademark of Intel Corporation.

™ PathWay is a trademark of The Wollongong Group, Inc.

.

™ OSF/1 is a registered trademark of the Open Software Foundation., Inc.

™ The DIGITAL Logo, CI, DECmessageQ, DECwindows, DECnet, Digital, MicroVAX, OpenVMS, TK, ULTRIX, VAX, VAXcluster, VAXft, VAXserver, VAXstation, and VMS are trademarks of Digital Equipment Corporation.