Software Product Description

PRODUCT NAME: DECnet/OSI Version 7.0 for OpenVMS VAX **SPD 25.03.39**

DESCRIPTION

DECnet/OSI Version 7.0 for OpenVMS VAX is an implementation of Phase V of the Digital Network Architecture (DNA) for the OpenVMS operating system for VAX hardware. DECnet/OSI integrates DECnet and Open Systems Interconnection (OSI) network protocols, allowing both stacks to share integrated network functions up to the Transport layer. Upper layers have been implemented as separate "towers," allowing existing DECnet and OSI applications to share the integrated Transport layer. Existing DECnet Phase IV and new DECnet and OSI applications are supported by DECnet/OSI. In combination with TCP/IP protocol stacks, OpenVMS systems can participate in multivendor, multiprotocol networks adhering to open networking standards.

DECnet/OSI for OpenVMS features include:

- Continued support for 32–bit network applications.
- X.25 network management interface, providing more efficient wide area network (WAN) management.
- The ability to configure all members of a cluster from any cluster member.
- The ability to generate an OpenVMS audit event when a connection is established.
- OSI application programming interfaces, formerly offered as a separate product.
- A new OSI Session layer programming interface (SPI).
- The ability to run DECnet and OSI applications over TCP/IP transports.
- Expanded naming options, allowing the use of a new and larger local namespace, DECdns, and/or DNS /BIND as naming services.
- The Graphical Network Management user interface for improved network management.
- Routing segregated mode to allow the Routing layer to choose a Phase IV router for those packets having a Phase IV compatible address. (Packets having a Phase V extended address are sent to a Phase V router by default.) This is a configurable option.
- Reverse path caching to capture path information for later use in reaching remote systems.
- Increased network size, supported through the use of ISO OSI addressing.
- Simplified installation and configuration process.
- The Network Control Language (NCL). Network management is based on DNA CMIP, Digital's implementation of the OSI international standard Common Management Information Protocol.
- An NCP emulator to facilitate installation of layered products written for DECnet Phase IV.
- Installation using the POLYCENTER Software Installation utility.
- Network naming, allowing the use of either local naming or fully distributed name services.
- Support for topologies using multicircuit and multihomed end nodes.
- Dynamic connections over X.25 networks.
- OSI implementation in accordance with current U.S. and U.K. GOSIP requirements:

  — Application, Presentation, and Session layers

  . File Transfer, Access, and Management (FTAM) application

  . Virtual Terminal application

  . Applicaton Service Elements (ASEs), including ACSE (Association Control Service Element)

  — Transport layer, classes 0, 2, and 4

— Lower layers

- OSI addressing formats, supporting very large network topologies
- End system to intermediate system (ES-IS) routing
- Connectionless Mode Network Service (CLNS) over local area network (LAN), wide area network (WAN), and X.25
- Logical Link Control type 2 (LLC2) for Connection–Oriented Network Service (CONS) over LAN
- Data Link layer, supporting High–level Data Link Control (HDLC) for wide area communications, ISO 8802-3 (Ethernet CSMA/CD) and FDDI LANs, and DDCMP for backwards compatibility. HDLC support includes the LAPB protocol for X.25 communications.
- Physical layer, with CSMA/CD, HDLC, and FDDI devices supported.

DECnet/OSI for OpenVMS is available in two forms: End System and Extended Function. Extended Function provides all the features of End System plus the OSI application gateways (FTAM - DAP gateway, VT-Telnet, VT-LAT), the DECdns Server, the OSI application programming interfaces and cluster alias. See the Software Licensing section for more details.

DECnet/OSI for OpenVMS offers task-to-task communications, file management, downline system and task loading, network command terminals, and network resource sharing capabilities using DNA, ISO and TCP/IP protocols. DECnet/OSI for OpenVMS communicates with adjacent and non-adjacent DECnet Phase IV, PATHWORKS, DECnet/OSI implementations on other OpenVMS, ULTRIX, and Digital UNIX systems, as well as OpenVMS systems running TCP/IP transports.

OpenVMS programs written in native mode programming languages can use DECnet/OSI for OpenVMS capabilities.

Depending on the system configuration, networks combining DECnet/OSI for OpenVMS systems with other DECnet, OSI and TCP/IP products may limit the functions available if all products do not support equal features.

Data Link Layer

DECnet/OSI for OpenVMS uses asynchronous, Ethernet, and FDDI communications controllers to interface with other network nodes.

LAN connectivity is provided by the CSMA/CD and FDDI controllers and drivers supporting ISO 8802-2 Logical Link Control (LLC) type 1 connectionless service and ISO 8802-3 LLC Type 2. DECnet/OSI also supports Ethernet V2 packets on CSMA/CD devices.

Use of FDDI packets larger than 1500 bytes requires a Phase V router on the FDDI LAN. As with cluster alias support, you may configure the Phase V router to run the Phase IV distance vector routing protocol or the Phase V link state routing protocol.

WAN connectivity is provided by optional X.25 software supporting host-based synchronous communications interfaces.

Supported synchronous devices allow for the use of DDCMP, DEC-HDLC, LAPB/E and SDLC protocols. BISYNC and GENBYTE protocols are also supported on some options. WAN device drivers are required by X.25 to establish host-based wide area connections.

Synchronous controllers use DDCMP or HDLC, either when directly connected or when connected via modems, to provide full- or half-duplex communications over point-to-point lines. Synchronous DDCMP multipoint tributary connections are also supported. Asynchronous controllers use DDCMP, either when directly connected or when connected via modems, to provide only full duplex communications over point-to-point lines. Error correcting and data suppression modems are not supported.

Asynchronous lines are supported only to other systems running DECnet/OSI for OpenVMS VAX, DECnet–VAX, DECnet–RSX, and DECnet–DOS.

DDCMP operation is not supported in cases where an asynchronous physical communications line is emulated by lower-level protocols or communications subsystems. Examples of this include X.29 virtual terminals, asynchronous connections as emulated by terminal servers, and connections via data switches.

DECnet/OSI for OpenVMS allows up to four circuits to be defined and operational on an end system. This capability allows a single end system to be connected to up to four separate LANs or WANs. Digital recommends that the circuits be equal in capacity and connectivity.

X.25

The DECnet/OSI for OpenVMS VAX license includes the right to use X.25 Access software, formerly known as VAX P.S.I. Access. The right to use X.25 Native mode software, formerly known as VAX P.S.I., requires an additional license. See the Optional Software section of this SPD for ordering information. The X.25 software in DECnet/OSI for OpenVMS is backwards compatible with systems running the older VAX P.S.I. products. Applications written to the older P.S.I. interfaces will continue to run, unchanged, with the X.25 software supported on DECnet/OSI.

The X.25 native software allows suitably configured DECnet/OSI systems to connect to Packet Switching Data Networks (PSDNs) conforming to CCITT recommendation X.25 (1980 or 1984) and/or to International Standards (ISO) 7776 and 8208. This allows a DECnet/OSI system to function as data terminal equipment (DTE) or be addressed as data circuit terminating equipment (DCE) as follows:

- A packet mode DTE connected to a supported PSDN (supported PSDNs are listed in Table 4) conforming to CCITT Recommendation X.25 (1980, 1984)
- A packet-mode DTE connected to a CSMA/CD LAN conforming to ISO 8802-3 using ISO Logical Link Control Type 2 (LLC2) as specified in ISO 8881
- A packet-mode DTE/DCE connected to a DCE/DTE conforming to ISO 7776 and 8208

X.25 can be configured for native operation to support direct connections from a VAX system to one or more PSDNs, each of which may have one or more DTEs. It also allows communication with any non-Digital X.25 system on the same LAN that supports the ISO Logical Link Control Type 2 (LLC2) protocol.

X.25 Access Software

The X.25 access software uses a connector node to provide the physical connection to a PSDN. A connector node can be an X25router 100/2000 (SPD 28.86.xx), a DEC X25gateway 100/500 (SPD 32.97.xx), DECnis router (SPD 36.95.xx), among others.

DECnet/OSI logical links are established by OpenVMS to connect the X.25 Access system to the X.25 Connector system. These links may use any supported DECnet/OSI communications path between the X.25 Access system and the Connector system, provided they themselves do not use an X.25 connection. X.25 Access uses these links to transmit X.25 or X.29 messages between the Connector system and the X.25 Access system.

Note that successful use of an X.29 connection, either for incoming X.29 calls or by means of the host-based PAD utility supplied with X.25 Access, depends on the DECnet/OSI circuit between the Access and Connector system having a low response time. This requires high-speed DECnet/OSI connections and few intermediate routing systems.

A single X.25 Access system can connect to one or more Connector systems concurrently, and hence access all PSDN(s) accessible from those Connector systems. When X.25 Access is used, one of these Connector systems/PSDN combinations must be established as the default gateway for outbound and inbound traffic.

Features

The X.25 software provides the Connection-Oriented Network Service (CONS) to allow mapping between a destination NSAP address and a destination DTE address according to ISO 8348.

The X.25 software can be used in the following ways:

- Process-to-process (X.25) communications
- Process-to-terminal (X.29) communications
- Terminal-to-process (X.29) communications

Process-to-Process (X.25) Communications

The X.25 software provides access for OpenVMS user programs written in VAX MACRO and high-level languages such as VAX FORTRAN. The X.25 software does not support programs executing in PDP–11 compatibility mode.

The X.25 programming interface allows application programs to access the service interface to the packet level by means of the standard OpenVMS QIO interface. Functions include setting up and breaking network connections, data transmission and reception, sending and receiving interrupt messages, and resetting virtual circuits. The interface allows splitting and recombining of messages that are longer than the packet size selected for the circuit.

The X.25 programming interface enables an application program using OpenVMS system services to communicate with complementary software on other systems (Digital or non-Digital DTEs) accessible through the X.25 connection.

Process-to-Terminal and Terminal-to-Process (X.29) Communications

The X.25 software supports terminal communications according to CCITT recommendations X.3, X.28, and X.29 for outgoing and incoming terminal traffic

X.25 native users may make outgoing calls to other Digital or non-Digital CPUs, or suitable network PADs accessible through a PSDN, using the X.29 programming interface or by using the host-based packet assembly /disassembly (PAD). The PAD facility accepts DCL-style commands or CCITT X.28 commands.

The X.29 interactive terminal interface allows remote asynchronous terminals (character-mode DTEs) connected to the network to communicate with the X.25 host in a manner similar to local terminals. Dissimilarities may appear as a result of delays caused by the PSDN(s) and the various PAD parameter settings chosen by the user or peculiar to the PSDN(s). In this case, the remote PAD must perform code conversions between ASCII and the code the terminal uses. Note that there may be some restrictions imposed by the PSDN(s).

Terminal parameters defined in the X.3 recommendation are supported explicitly. Network-specific enhancements or extensions to the X.3 parameters are available at both the X.29 programming interface and the host-based PAD user interface. Terminal processes that depend on these extensions may cause problems when used on other PSDNs or when accessing one PSDN via another (for example, international access).

The X.29 terminal interface is compatible with the local terminal interface, except for those facilities where the PSDN does not provide adequate support. The X.29 interface includes support for specific X.29 signaling.

The host-based PAD employs a DCL-style command syntax, offering the ability to reference the CCITT PAD parameters symbolically, as well as an interface that conforms to the standard CCITT command signals. The host-based PAD also offers the capability of running multiple X.29 sessions using the OpenVMS SPAWN and ATTACH commands. The host-based PAD supports reception of reselection messages to allow X.29 calls to be redirected to another DTE.

The X.29 programming interface is similar to the X.25 one and allows an OpenVMS process to control the virtual circuit (for example, to modify PAD parameters or clear a call). The X.29 programming interface can also be used to make an outgoing call to a remote PAD. The programming interface allows generation of PAD reselection messages.

The maximum number of terminals supported (both local and remote X.29) must not exceed the number supported by OpenVMS. The use of PVCs for X.29 connections is not defined by the 1980, 1984 and 1988 CCITT Recommendations and is not supported.

Static X.25 Circuits

X.25 allows the use of DECnet/OSI facilities over static X.25 circuits. These circuits are called statically assigned circuits.

Virtual Circuits

X.25 native software supports both permanent virtual circuits (PVCs) and switched virtual circuits (SVCs). One virtual circuit is used for each:

- Incoming or outgoing X.29 terminal connection
- X.25 connection, using either a SVC or PVC

X.25 native supports a maximum of 4,096 virtual circuits per system.

X.25 Access supports a maximum of 900 virtual circuits per system.

Note that each virtual circuit requires a certain amount of memory so the number of logical virtual circuits possible on any single system is limited by the available memory.

Configuration

The X.25 software includes a configuration procedure to configure a VAX system. This menu-based interface allows network managers to create an initialization script to configure entities and their attributes after initial installation.

X.25 Accounting

The X.25 Accounting utility enables the system manager to obtain information PSDN usage. X.25 Accounting can extract the information collected and display the information directly on a terminal or write the information to a file. It can also extract information for selected records, or produce a summary. The X.25 Accounting utility is compatible with the OpenVMS Accounting utility.

In the case of incoming X.29 calls, no information about the process/account on to which a user is logged can be stored. However this information may be obtained by correlation with OpenVMS accounting records.

X.25 Security

X.25 security allows control of both remote DTE access to the X.25 software, and access to the PSDN(s) by OpenVMS processes. X.25 security allows for access control on the basis of:

- The remote DTE numbers involved, either on the basis of individual DTE numbers or groups of DTE numbers with common leading digits.
- The process making the outgoing call or the application handling the incoming call.

X.25 Mail

The X.25 Mail utility allows communications via electronic mail between two OpenVMS systems running the X.25 software in native or in access mode. Refer to the OpenVMS Operating System SPD 25.01.xx for more information about OpenVMS Mail.

Synchronous Communications Devices

The X.25 native software supports the synchronous communications devices listed in Table 2. The devices are different in many respects, particularly in the amount of CPU used. For optimum performance, Digital recommends the use of a DMA device.

Line Interfaces

The electrical line interface used for PSDN communications depends on the hardware device used. X.25 native provides only the electrical interface of a DTE rather than a DCE, and does not provide clock signals. Therefore, an external clock source (for example, a modem) is required.

The range of supported synchronous communications devices covers V.24, V.35 and V.10, RS232, RS422 and RS423 interfaces over point-to-point, 4-wire, synchronous, full-duplex lines at transmission rates up to 256K bits per second.

Prerequisite Support

For new PSDNs, the customer and Digital must jointly prepare a Network Profile and Customer Support Plan covering all the intended network systems, their usage of SVCs, PVCs, and other network facilities, and their support. Without this Network Profile and Customer Support Plan, Digital cannot support the network connections.

Optional Facility Support

Support for any given CCITT X.25 facility depends on support for it in the appropriate supported PSDN. Generally, the X.25 software is designed to disallow use of facilities that the PSDN will not support. Table 1 shows what X.25 is capable doing, rather than what it will do when used on any particular PSDN. The product documentation describes specific facility availability for the supported public PSDNs.

Table 1
Optional X.25 Facility Support

Option	Paragraph Number in CCITT X.25 (1984)	DECnet/OSI Support
Extended packet sequence numbering	6.2	Yes
Non-standard default window size	6.10	Yes
Default throughput classes assignment	6.11	N/A
Packet retransmission	6.4	See Note[1]
Incoming calls barred	6.5	N/A
Outgoing calls barred	6.6	N/A
One-way logical channel outgoing	6.7	N/A
One-way logical channel incoming	6.8	Yes
Closed user group (CUG)	6.14.1	Yes
CUG with outgoing access	6.14.2	Yes
CUG with incoming access	6.14.3	Yes
Incoming calls barred within CUG	6.14.4	Yes
Outgoing calls barred within CUG	6.14.5	Yes
CUG selection	6.14.6	Yes
CUG with outgoing access selection	6.14.7	No
Absence of both CUG selection facilities	6.14.8	Yes
Bilateral CUG (BCUG)	6.15.1	Yes
BCUG with outgoing access	6.15.2	Yes
BCUG selection	6.15.3	Yes
Reverse charging	6.18	Yes
Reverse charging acceptance	6.19	Yes
Local charge prevention	6.20	N/A
Network user identification	6.21	Yes
Charging information	6.22	Yes
RPOA selection	6.23	Yes
Hunt group	6.24	See Note[2]
Call redirection	6.25	N/A
Called line address modified notification	6.26	Yes

[1] This is not applicable to a DTE. The X.25 software does not support receiving a level 3 Reject packet and does not generate one.

[2] The individual DTEs must be assigned their own addresses independent of the hunt group address. X.25 can then ignore the hunt group. Alternatively, each line on the same CPU can be configured as if connected to a separate PSDN; each with the same DTE address.

Table 1 (Cont.)
Optional X.25 Facility Support

Option	Paragraph Number in CCITT X.25 (1984)	DECnet/OSI Support
Call redirection notification	6.27	Yes
Transit delay selection and indication	6.28	Yes
On-line facility registration	6.1	No
Non-standard packet size	6.9	Yes
Flow control parameter negotiation	6.12	Yes
Throughput class negotiation	6.13	Yes
Fast select	6.16	Yes
Fast select acceptance	6.17	Yes
D-bit modification	6.3	No
Abbreviated address	-	N/A
Datagram queue length selection	-	No
Datagram service signal logical channel	-	No
Datagram non-delivery indication	-	No
Datagram delivery confirmation	-	No
Call statistics (non-X.25)	-	See Note [3]

[3] The X.25 software does not generate these statistics, however, incoming statistics supplied by the PSDN are available in the accounting records and at the programming interface.

The X.25 software does not support the following items of the CCITT X.25 recommendation:

- Multi-link protocol
- Use of the D-bit in outgoing packets

Wide Area Network (WAN) Device Drivers

The WAN device drivers included in DECnet/OSI for OpenVMS VAX support synchronous communications. The device drivers all offer a supported user ($QIO) interface.

The device drivers all support full-duplex and half-duplex operation, where appropriate to the protocol. DDCMP and SDLC device drivers all support both point-to-point and multipoint tributary operation. ‘BISYNC’ refers to the IBM® implementation of bisynchronous protocol. The GENBYTE protocol is a general byte-oriented protocol for use with user-written framing routines on some of the device drivers.

The maximum permitted line speed depends on the line interface standard used and must not exceed 19.2 Kb/s for an RS232 /V.24 standard interface. There may be additional line speed restrictions for other line interface standards. Refer to the Software Product Descriptions for the synchronous interfaces for more information. The Digital communications layered products may not necessarily support operation of the synchronous interface at the maximum speed supported by the device driver. Refer to the appropriate Software Product Description for the Digital communications layered product for precise details of supported configurations.

In all cases, the data throughput over the synchronous lines is dependent on the user applications and system environment.

The device drivers all support modem control signal operation. Refer to the product descriptions for the appropriate synchronous interface for details of the modem signals supported by an individual device.

Supported controllers are listed in Table 2.

WAN device drivers include a pseudodriver (WANDRIVER) that provides a programming interface to the Data Link level for the LAPB, DEC-HDLC, and DDCMP protocols.

Network Layer

DECnet/OSI for OpenVMS supports both the Connectionless-Mode Network Service (CLNS) and the Connection-Oriented Network Service (CONS). DECnet/OSI supports end-system routing only.

Exchange of routing information between end systems and routers uses the ISO 9542 ES-IS routing protocol. This allows DECnet/OSI systems to autoconfigure as end systems with ISO 9542-conformant routers.

Addresses conform to the ISO 8348 Addendum 2 specification, allowing the support of large network topologies. As long as the system address stays within the addressing range of Phase IV systems (up to 1,023 systems per area and up to 63 areas per network), and uses the same initial domain part (IDP), Phase IV or Phase V routers may be used.

The Network layer supports the capability of an end system to be multicircuited and multihomed. Multicircuit support allows multiple circuits to be active simultaneously. This increases network reliability and data throughput. Multihomed end-system support allows a system to have up to three network entity titles.

Segregated routing mode is a settable attribute. It directs routing to choose a Phase IV router for those packets having a destination address that can be translated to the Phase IV format. All other packets will be sent to a Phase V router, if available.

The Routing layer is able to cache information about the paths that are used to reach remote nodes.

Transport Layer

DECnet/OSI for OpenVMS provides support for:

- OSI Transport protocol as specified in ISO 8073
- RFC 1006 and 1006 Extensions (Internet Draft) to allow DECnet and OSI applications to run over TCP/IP
- Network Services Protocol (NSP)

NSP, RFC 1006 and OSI transports support communications between DECnet, TCP/IP and OSI systems. NSP provides backwards compatibility with Phase IV DECnet systems.

RFC 1006 Extensions support for DECnet applications is provided via a kernel interface that has been implemented on all TCP/IP stacks available for use on the OpenVMS operating system. The supported applications include all licensed DECnet applications as well as layered products, and user-written applications that conform to the documented DECnet programming interfaces.

The IETF RFC 1006 is a specification for running OSI applications over TCP/IP. Operation of the FTAM and Virtual Terminal applications over a TCP/IP network is supported along with other layered OSI applications such as X.400 and X.500.

A separate TCP/IP stack is required on the same system with DECnet/OSI. See the Optional Software section of this SPD for information on supported TCP/IP products.

OSI Transport supports Transport Classes 0, 2, and 4 on connection-oriented networks and Class 4 on connectionless networks.

OSI Transport uses two types of network service:

- The Connectionless-Mode Network Service (CLNS) using the Internet protocol (ISO 8473) and ES-IS protocol (ISO 9542) to communicate across linked subnetworks. The inactive subset of ISO 8473 (null internet) is also supported for communications over a single ISO 8802-3 local area network.
- The Connection-Oriented Network Service (CONS).

Upper Layers

DECnet/OSI for OpenVMS software provides the OSI upper layer stack consisting of Session, Presentation and Application layers. The Application layer provides Association Control Service Element (ACSE); File Transfer, Management, and Access (FTAM); and Virtual Terminal (VT).

DECnet/OSI for OpenVMS also provides a range of DECnet applications and services including file and record access, remote terminal access, mail, and phone.

Applications

Transport Options for Applications

Applications written to the DECnet upper layers can now be run over NSP, OSI or TCP/IP transports via RFC 1006. This includes the network applications that are licensed with DECnet/OSI as well as user-written applications that adhere to the documented DECnet programming interfaces. The use of TCP/IP transports via RFC 1006 does not require any modification to the existing application.

Applications written to the OSI upper layers can now be run over OSI or TCP/IP transports.

Remote File Transfer

DECnet/OSI for OpenVMS supports two upper layer protocols for remote file transfer: the OSI Protocol File Transfer, Access and Management (FTAM) and the DECnet Data Access Protocol (DAP). *FTAM*

FTAM supports file transfer, access, and management between a DECnet/OSI for OpenVMS system and other systems with software adhering to ISO 8571. In addition, FTAM is conformant with NIST Phase II and Phase III agreements and is certified as being conformant to the released specifications of U.S. GOSIP, U.K. GOSIP and ENV41204.

In addition, FTAM:

- Provides users the ability to create, delete, rename, view, and copy files using DCL commands.
- Is implemented as an Application Service Element (ASE) of the OSI Application layer.
- Acts as the initiator or as the responder in a connection.
- Accesses and transfers files with both binary and character data. FTAM-1, FTAM-2, FTAM-3, and NBS-9 document types are supported.

DECnet/OSI for OpenVMS also supports gateway services between FTAM and DAP.

A full description of the FTAM services in DECnet/OSI for OpenVMS is provided in Appendix A of this document.

DAP

The DECnet Data Access Protocol (DAP) supports task-to-task communications, file and record access, and proxy access.

Task-to-Task Communications: For most applications, task-to-task communications can be programmed in a transparent manner where the remote task is treated as a full-duplex, record-oriented device. Transparent operation is provided with the following interfaces: system Service Calls, RMS calls (OPEN, GET, PUT, and CLOSE), and high-level language I/O statements (which are mapped to RMS calls). A nontransparent mode of task-to-task communications is offered by means of the system service interface that extends the capabilities provided by the transparent mode. These capabilities include support for interrupt messages and multiple inbound connect requests.

Using DECnet/OSI for OpenVMS, an OpenVMS program written in a native mode programming language can exchange messages with other user programs.

File Access: File access is supported to and from remote DECnet/OSI for OpenVMS systems, transparent to native mode high-level language programs using RMS. User programs can sequentially read, create, and delete files on a remote node.

Record Access: User programs can perform record level operations such as GET, PUT, UPDATE, DELETE, FIND, and REWIND to access and modify files residing on a remote OpenVMS node. In addition to sequential access to a file, several other access methods are supported through RMS using DECnet/OSI for OpenVMS. These methods include random access by relative record number, random access by key value, random access by record file address (RFA), and block I/O access by virtual block number.

Proxy Access: Remote users can have access to up to 15 proxy accounts on a specific remote system. One proxy account should be designated as the default proxy account on the remote system.

Command Language File Management: Most OpenVMS Digital Command Language (DCL) commands can be used to perform remote file operations. These commands include: ANALYZE, APPEND, BACKUP, CLOSE, CONVERT, COPY, CREATE, DELETE, DIFFERENCES, DIRECTORY, DUMP, OPEN, PRINT, PURGE, READ, SEARCH, SUBMIT, TYPE, and WRITE. The operation of these commands is transparent except for commands that invoke processing on a specific system (for example, SUBMIT/REMOTE and PRINT /REMOTE). Only a node name added to a file specification is required to invoke the network capabilities via one of these commands.

Using the COPY command, a user can transfer sequential, relative, and indexed-sequential (ISAM) files between DECnet/OSI for OpenVMS nodes that support compatible file structures and record formats. Users can transfer sequential or relative files with fixed length, variable length, or variable length with fixed control field records between two DECnet/OSI for OpenVMS systems. Similarly, multikeyed indexed files with variable or fixed length records are supported.

The SUBMIT/REMOTE command allows command files residing on a remote node to be submitted for execution at the remote node. The command file must be in the format expected by the node responsible for execution. DECnet/OSI for OpenVMS allows OpenVMS command files to be received from other systems and executed.

The DCL command, EXCHANGE/NETWORK, allows the transfer of files to or from heterogeneous systems. This command gives users the option to transfer file types between MS–DOS®, ULTRIX, and UNIX systems and OpenVMS systems regardless of record semantics. Unlike the COPY command, which preserves file and record organization during a file transfer, this command enables the user to modify file and record attributes during file transfer.

OSI Application Programming Interfaces

The OSI application programming interfaces enable users to write distributed applications that communicate over open networks and use the OSI services provided by DECnet/OSI.

The interfaces are provided under the Extended Function license and include:

- An interface to FTAM (File Transfer, Access, and Management)
- Interfaces to the ACSE (Association Control Service Element) and Presentation layers
- An interface to ROSE (Remote Operations Service Element)
- An interface to the OSI Session Layer

The interfaces permit application writers to use the services of the OSI upper layers in their applications.

The APIs allow for the development of applications on DECnet/OSI systems. Source code must be compiled and linked with the APIs. The APIs are required on the development system and the compile/link system. The resulting application can be run on any DECnet/OSI system with the appropriate operating system. The APIs are not required on these target systems. See Appendix C for details on the OSI APIs.

Network Virtual Terminal

DECnet/OSI supports two upper layer protocols for terminal access: the OSI Virtual Terminal protocol and the DECnet Command Terminal protocol.

VT

Virtual Terminal (VT) supports the ISO Virtual Terminal Protocol (ISO 9041). This protocol allows remote logins and access to remote applications between DECnet/OSI for OpenVMS systems and any remote system, including multivendor systems, that also run an ISO-compliant Virtual Terminal implementation.

Virtual Terminal is implemented as an Application Service Element (ASE) of the OSI Application layer.

Virtual Terminal may act as the terminal/initiator (for a local user) or as the host/responder (for the remote user).

A full description of Virtual Terminal features is provided in Appendix B of this SPD. *SET HOST*

The DCL command SET HOST allows a terminal user on one DECnet/OSI for OpenVMS node to establish a logical connection to another DECnet/OSI node that uses the Command Terminal (CTERM) protocol. This connection makes the terminal appear to be physically connected to the remote system and the operator can use all the standard system and network utilities supported by that remote node. This capability is particularly useful for doing remote program development and allows the terminal users on smaller application-oriented systems to use the resources of larger development-oriented systems.

Other interfaces are standard parts of DECnet/OSI for OpenVMS. Users can develop programs and procedures based upon these interfaces for such functions as file access and task-to-task communications on individual systems. Because the DECnet/OSI for OpenVMS interfaces stay the same, the programs and procedures developed on an individual system can be used in a network environment without being modified.

Services

Downline Loading

DECnet/OSI for OpenVMS allows for the loading of an unattended system using the services provided by the Maintenance Operations Module (MOM). MOM provides a set of maintenance operations over various types of circuits by using the Maintenance Operations Protocol (MOP). A loadable system is a system that has a load device enabled for MOP service functions and for which a properly formatted load file is supplied. Downline loading involves transferring a copy of the load file image to a remote target node. Load requests can come from the local DECnet/OSI for OpenVMS operator or from the target node. Downline loading is supported for Digital server products. However, this facility is not supported over asynchronous DECnet/OSI for OpenVMS VAX connections.

Downline Task Loading

Initial task images for loadable systems can be stored on OpenVMS file system devices and loaded into remote nodes. Programs already executing on loadable systems can be checkpointed to the host OpenVMS file system and later restored to main memory in the node. These features simplify the operation of network systems that do not have mass storage devices.

Downline task loading is not supported over asynchronous DECnet/OSI for OpenVMS VAX connections.

Upline Dumping

Memory images of adjacent nodes connected by DECnet/OSI for OpenVMS can be written or dumped into a file on an OpenVMS system. This facility provides assistance in troubleshooting in the event of a system crash. This facility is also supported for Digital server products.

This facility is not supported over asynchronous DECnet/OSI for OpenVMS VAX connections.

Mail

The OpenVMS Mail utility allows transmission of text messages between users on systems supporting MAIL-11. The DECnet/OSI for OpenVMS software allows users to exchange mail with users of other DECnet/OSI systems.

Phone

The OpenVMS Phone utility allows users to send and receive data interactively from one user's terminal to another user's terminal. DECnet/OSI allows users on different systems in the same DECnet/OSI network to exchange information.

VMScluster Alias

DECnet/OSI for OpenVMS supports the ability to access nodes in a VMScluster using a separate alias node address, while retaining the ability to address each node in the cluster individually. Not all network objects may be accessed using this mechanism. The maximum number of nodes supported for a cluster alias is 94. Refer to the VMScluster Software Product Description (SPD 29.78.xx) for relevant restrictions.

DECnet/OSI no longer requires a cluster member to be configured as a router. Clusters in a DECnet/OSI environment require a reachable IS-IS compliant router on the LAN.

Network Management

Network management is provided with the Network Control Language (NCL). Network management implements the DECnet/OSI layered model, based on the Digital hierarchical structure called Enterprise Management Architecture (EMA).

Users can access NCL through either a command line interface or a graphical user interface (GUI). The GUI allows network managers to view the status of network components and control those components from a Motif-based window interface.

The DECnet/OSI for OpenVMS network management software allows system and network managers to:

- Control and monitor the operation of a network and provide information related to network traffic and performance
- Configure network operating parameters
- Start up and shut down network components as needed
- Detect and isolate network problems, and return the network to service once repaired.

In addition, network management can provide information, warning network managers of faulty or failing network components, both hardware and software.

Network Command Language (NCL) is provided as a utility to the network manager to perform the operations described above.

Network managers can also use NCL to test specific components of the network. NCL enables transmission and reception of test messages either between systems or through controller loopback arrangements. The messages can then be compared for possible errors. NCL helps users isolate network problems.

DECnet/OSI for OpenVMS provides network event logging to a terminal device, disk file, or remote system. NCL users can enable and disable the event logging facility as well as optionally filter specific events.

NCL uses the DNA Common Management Information Protocol (CMIP) which permits entity management from a single location anywhere in the DECnet/OSI network.

The Common Trace Facility (CTF) allows the network manager to collect and display information about specific protocol exchanges between systems.

DECnet/OSI for OpenVMS supports an ISO CMISE application programming interface (API) conforming to the Service Definitions in ISO 9595. The API allows for development of applications that can communicate with other management applications conforming to ISO 9595 on remote nodes in the network.

DECnet/OSI supports the DECnet Phase IV NCP for remote management of Phase IV DECnet systems.

Name Service Options

DECnet/OSI for OpenVMS allows the use of one or more naming services. The available services are DECdns, DNS/BIND and a new local namespace. Node name and addressing information is stored in the native name service; TCP/IP information is maintained in DNS /BIND and DECnet and OSI information is maintained in DECdns or the local namespace.

When more than one name service is being used, a configurable search list defines the order in which the existing services are to be accessed.

Local Namespace

The local namespace replaces the Local Name Option (LNO). Using the local namespace, up to 100,000 nodes can be defined in a local naming database. A migration tool is available to move Phase IV and/or LNO databases to the new large local file format and /or DECdns format.

DNS/BIND

DECnet can now retrieve TCP/IP naming and addressing information from the DNS/BIND name service.

DECdns

DECnet/OSI for OpenVMS provides a global naming service with the Digital Distributed Naming Service (DECdns).

The full DECdns service provides a consistent, networkwide set of names for network resources called the namespace. This namespace is maintained by one or more DECdns server systems. Digital recommends that DECdns servers be installed on at least two systems in every LAN. This should provide adequate service and redundancy.

The features provided by DECdns include:

- A networkwide name-to-attribute mapping service which allows selected Digital applications to create, read, modify, and delete names in the namespace
- A hierarchical structure permitting a large number of names to be stored and distributed across the network
- Access control to each name in the namespace
- Management and event logging

Distributed Time Service

DECnet/OSI for OpenVMS provides a network time service with DECdts, the Digital Distributed Time Service. DECdts provides precise, fault-tolerant clock synchronization for systems in a LAN or WAN. Time is provided in Coordinated Universal Time (UTC) and may be used across a global network. Several forms of time provider are supported, and a callable interface for applications allows users to add their own time providers. DECdts may be used by distributed applications to determine event sequencing, duration, and scheduling.

DECnet/OSI for OpenVMS Operation

DECnet/OSI is implemented under OpenVMS as an Ancillary Control Process (ACP) and a network device driver with Digital-supplied executive-level components and user-level programs.

The normal OpenVMS protection has been incorporated in the operation of DECnet/OSI. For example, incoming connects, including file access and file transfer requests, are protected by the normal OpenVMS login and file protection mechanisms. Outgoing connects, including file access and file transfer requests, can include user password information that is implicitly specified via NCL, or explicitly specified by the user for verification on the remote node.

DECnet/OSI for OpenVMS Configuration and Performance

DECnet/OSI can be configured using either the BASIC or ADVANCED configuration options. The BASIC option is the default. It uses existing system defaults and requires minimal information from the system manager. The ADVANCED option should be used where specific customization is required.

With DECnet Version 6.3 and later, cluster nodes are configurable from any node within the cluster.

As with any network protocol, the performance of a given DECnet/OSI for OpenVMS node is a function not only of the expected network traffic and resultant processing, but also of the amount of concurrent processing specific to that node. Thus, node performance depends on many factors including:

- CPU type
- Number and type of devices attached to the particular CPU
- Number of device interrupts per unit time
- Communications line(s) characteristics
- Number and size of buffers
- Message size and frequency of transmission
- Applications in use

It is important to note that the rate at which user data can be transmitted (throughput) over a communications line can sometimes approach, but will never exceed, the actual line speed. This is because the actual throughput is a function of many factors, including the line quality, protocol overhead, topology, and network application(s), as well as the factors cited in this section.

The performance of DECnet/OSI is comparable to the performance of DECnet Phase IV.

Standards Conformance

DECnet/OSI for OpenVMS has been designed and implemented to conform to the following standards:

- ISO
  - 4335
  - 7776, 7809
  - 8073, 8208, 8327, 8473, 8571, 8650, 8802-2, 8802-3, 8823, 8878, 8881
  - 9314, 9542, 9041
  - 3309
- EN
  - 41 204
  - 41 205
  - 41 206
  - 41 207
- CCITT Recommendation X.25 (1978, 1980, or 1984) using the LAPB or LAPBE variants of the X.25 packet level and data link protocols
- U.S. GOSIP V2.0
- U.K. GOSIP V4.0

Contact your local Digital office for the most recent conformance certificates.

Documentation

The documentation for DECnet/OSI for OpenVMS is supplied in four parts:

- The core documentation set shipped with the H-kit
- An optional end user guide (one copy is shipped with the core documentation)
- An optional programming guide describing $IPC, $QIO, OSI Transport, X.25, WANDD and DECdts programming interfaces
- An optional X.25 documentation set covering accounting, X.29 and X.25 Mail

New features are extensively documented in the Release Notes.

The documentation for DECnet/OSI for OpenVMS is shipped on the OpenVMS layered product CD-ROM. Hard copy of the documentation is available as a separate order.

INSTALLATION

DECnet/OSI software is customer installable. Installation services are available for customers who desire installation of the software product by an experienced Digital Software Specialist.

Digital requires that a customer's first use of X.25 include Digital Installation Services. These services provide for installation of the software product by an experienced Digital Software Specialist.

Customer Responsibilities

Before Digital Services can install the software, the customer must:

- Ensure that the system meets the minimum hardware and software requirements (as specified in the relevant SPDs)
- Prior to installing Digital hardware or software, obtain, install, and demonstrate as operational any modems and other necessary customer equipment or facilities to which Digital's communications hardware or software will connect
- Designate one adjacent node to verify installation /connectivity
- Make available for a reasonable period of time, as mutually agreed upon by Digital and the customer, all hardware communications facilities and terminals that are to be used during installation.

Delays caused by any failure to meet these responsibilities will be charged at the then-prevailing rate for time and materials.

Installation of DECnet/OSI for OpenVMS consists of the following tasks:

- Verify that all components of DECnet/OSI for OpenVMS have been received.
- Verify that the necessary versions of the OpenVMS software and documentation are available.
- Verify the appropriate SYSGEN parameters.
  Note: If a software specialist is required to modify the previously installed operating system parameters, a time and materials charge will apply.
- Create any necessary DECnet/OSI for OpenVMS accounts and directories.
- Enable software by registering the License Product Authorization Key (PAK) .
- Install the DECnet/OSI software on the target system using the POLYCENTER Software Installation (PCSI) utility .
- Verify the proper installation of DECnet/OSI for OpenVMS by running a series of tests to show connectivity to a designated node.

Connectivity to all other nodes within the network is the responsibility of the customer. Digital recommends the use of NCL to help verify connectivity.

In some cases, the PSDN supplier (or PTT) may impose restrictions, limitations, or requirements on the proposed Digital network configuration. The customer must understand and adhere to these controls for each and every network.

For installation of the X.25 software, the customer must:

- Demonstrate equivalence of operation for modems other than Bell 208A, 208B, 209, 212A synchronous modems, or in Europe, PTT approved modems.
- Subscribe to the Open User Group and to at least two Switched Virtual Circuits (SVCs) to complete the product's installation checkout (this tests loops information from the X.25 native system to the PSDN and back to X.25 native system). Systems that use Permanent Virtual Circuits (PVCs) or Closed User Groups only, or where the PSDN does not support calls to the same DTE address, require specially negotiated arrangements for Digital installation of the product.

HARDWARE REQUIREMENTS

Refer to the OpenVMS VAX Operating System Software Product Description (SPD 25.01.xx) for hardware requirements and processor support. Refer to the configuration charts listed in the OpenVMS VAX Operating System SPD.

For general device or controller descriptions, refer to the *Networks and Communications Buyer's Guide*.

Processors Supported for OpenVMS Systems:

VAX:	VAX 4000 Model 100 VAX 4000 Model 100a VAX 4000 Model 200 VAX 4000 Model 300 VAX 4000 Model 400 VAX 4000 Model 400a VAX 4000 Model 500 VAX 4000 Model 500a VAX 4000 Model 600 VAX 4000 Model 600a VAX 4000 Model 700a
	VAX 6000 Model 200, 300, 400 VAX 6000 Model 500, 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, Model 210, Model 300, VAX 9000 Model 400 Series
	VAX 10000 Model 600 Series
	VAXft Model 110/310/410/610/612
	VAX–11/730, VAX–11/750, VAX–11/780, VAX–11/785
MicroVAX :	MicroVAX II, MicroVAX 2000 MicroVAX 3100 Model 10/10E, 20/20E, Model 30/40, Model 80, Model 90 MicroVAX 3300, MicroVAX 3400, MicroVAX 3500, MicroVAX 3600,MicroVAX 3800 MicroVAX 3900
VAXstation :	VAXstation II, VAXstation 2000 VAXstation 3100 Model 30/40, Model 38/48, Model 76 VAXstation 3200, VAXstation 3500 VAXstation 3520, VAXstation 3540 VAXstation 4000 VLC, VAXstation 4000 Model 60, VAXstation 4000 Model 90
VAXserver :	VAXserver 3100 Model 10/10E, 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 210/220 VAXserver 6000 Model 310/320 VAXserver 6000 Model 410/420 VAXserver 6000 Model 510/520 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.

The RL02/R80 configuration of the VAX–11/730 is required.

Disk Space Requirements (Block Cluster Size = 1)

For OpenVMS VAX Systems

Disk space required for installation of base software: 65,000 blocks

Disk space required for installation of all optional software: 101,000 blocks

For WAN Device Drivers

Disk space required for installation: 5,000 blocks (2.5 MB)

The disk space required for use (permanent) depends on the number and type of drivers installed, but should not exceed the space required for installation.

For X.25 Software

Disk space required for installation: 8,000 blocks (4 MB)

Disk space required for use (permanent): 10,000 blocks (5 MB)

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

The WAN device drivers are fully supported when installed on any valid and licensed VMScluster configuration. A device driver may only be accessed from the system, or node, containing the communications device.

On common disks, in order to configure some product-specific attributes that must remain unique to each system, certain components of the product are stored in the system-specific environment (SYS$SPECIFIC:[SYSxxx]) of the VAX computer on which the installation is being performed. The remaining components of the product, including some or all of the executable images, will be installed common to all accessing VAX systems.

Though some or all of the product's executable images will exist in the common system directory (SYS$COMMON:[SYS$LDR]) and be executable by all VAX computers sharing this directory, it is essential that the product be installed separately on all VMScluster or Local Area VMScluster licensed VAX systems requiring access in order to uniquely configure the product for each.

The same version of WAN device drivers must be used on all systems in VMScluster systems accessing a common disk and in Local Area VMScluster configurations.

Communications Devices

One of the communications devices listed in Table 2 is required for use with X.25 native and WANDD.

Table 2

Synchronous Communications Devices

Busless Systems	Q–bus Systems	VAXBI Systems	VAXft Systems	UNIBUS Systems
DSH32	DPV11	DMB32	DSF32	DMF32
DST32	DSV11	DSB32		DUP11
DSW42				
DSW41				
DSW21				

Refer to Table 3 for detail of line speeds for supported protocols.

- The DSH32 (dual sync line) and the DST32 (single sync line) are multifunction devices for busless systems. The communications controllers support DMA transfers for bit-oriented protocols and perform character transmission, reception, and framing in microcode.
- The DSW21 and DSW41 (single line) and the DSW42 (dual line) are high-performance synchronous communications controllers for busless MicroVAX 3100 and VAXstation 4000 systems. The communications controllers support DMA transfers for bit- and character-oriented protocols. The communications controller performs character transmission, reception, and framing in microcode.
- The DPV11-DA or the DPV11-S (Q–bus) and DUP11 (UNIBUS) are character interrupt devices for data link handling. CPU cycles are required for data link level protocol processing and for each character sent and received.
- The DSV11 is a dual-line synchronous communications device for all Q–bus systems. The DSV11 supports DMA transfers for bit- and character-oriented protocols. The communications controller performs character transmission, reception, and framing in microcode.
- The DSF32 (MI-bus) is a multifunction device for VAXft fault tolerant systems. The DSF32 provides a dual-line synchronous capability and supports DMA transfers for bit- and character-oriented protocols. The communications controller performs character transmission, reception, and framing in microcode. The two line can be used independently to run different protocols or can be used to provide a failover capability between lines in the event of an application or system failure.
- The DMB32 (BI-bus) and the DMF32 (UNIBUS) are multifunction devices that include a single DMA line. The communications controllers perform character transmission, reception, and framing in microcode.
- The DSB32 (BI-bus) is a dual-line synchronous communications device for VAXBI systems. The DSB32 supports DMA transfers for bit- and character-oriented protocols. The communications controller performs character transmission, reception, and framing in microcode.

Notes:

- The DSH32 and DST32 are not supported on VAXstation 2000 and VAXstation 3100 systems.
- The DSB32 is not supported for use on the VAX 7000 or VAX 10000 systems.
- The DSB32 is not supported for use on a VAX 6600 with extended addressing (>512MB of physical memory) enabled.
- The DMB32, DMF32, and DSH32 are multifunction devices. Only the device driver for the synchronous ports are included in the DECnet/OSI kit.
- The DMB32 must be revision J or later for applications using GENBYTE, HDLC or SDLC protocols.
- The DMF32 must be revision J of later for applications using HDLC protocol and revision M is required for simultaneous operation of the printer port.
- The DMB32 only supports a synchronous line speed of 19.2 Kb/s if any asynchronous lines are in use. The DMB32 is also supported on VAX 7000 and VAX 10000 systems configured with extended addressing, (32-bit physical addresses).

- The DMF32 is supported on a VAXBI system fitted with a DWBUA UNIBUS adapter for DDCMP and GENBYTE protocol use only.
- The DSH32 running the DDCMP protocol only supports a single line at 9.6 Kb/s on a MicroVAX 2000. Running the BISYNC protocol it only supports a single line at 4.8 Kb/s on a MicroVAX 2000.
- The DSV11, DSW42, and the DSB32 support the use of different protocols on each of the two lines of the device.
- The DUP11 is supported on a VAXBI system fitted with a DWBUA UNIBUS adapter for VAX 8200, VAX 8250, VAX 8300, VAX 8300, VAX 8350, VAX 8500, VAX 8530, VAX 8550, VAX 8700, and VAX 8800 systems only.

All the WAN device drivers included with DECnet/OSI offer support for a user ($QIO) interface. Table 3 lists the device driver characteristics and indicates the maximum permitted line speeds in Kb/s for each of the line protocols supported by the individual drivers.

Table 3

Synchronous Device Driver Characteristics

Device	Code	Maximum Line Speed for Supported Protocols (Kb/s)		
		DDCMP	HDLC	SDLC
DMB32	SI	19.2	38.4	38.4
DSB32	SL	64/64	192/64	192/64
DSV11	SJ	256/64	256/64	256/64
DMF32	XG	19.2	19.2	19.2
DPV11	SE	9.6	9.6	9.6
DSH32	ZS	19.2	19.2	19.2
		/9.6	/9.6	/9.6
DST32	ZS	9.6	19.2	19.2
DUP11	SE	4.8	9.6	9.6
DSF32	SF/SM	64/64	64/64	64/64
DSW42	ZT	256/64	256/64	256/64
DSW41	ZT	256	256	256
DSW21	ZT	256	256	256

Note: Where two line speeds are shown, for example 256/64, the higher speed refers to single line operation of the device while the lower speed refers to dual line operation of the device.

The WAN device drivers included with DECnet/OSI also provide a pseudodriver (WANDRIVER) that provides a programming interface to the data link level for the LAPB, DEC-HDLC, and DDCMP protocols.

The device drivers all support full-duplex and half-duplex operation, where appropriate to the protocol. DDCMP and SDLC device drivers all support both point-to-point and Multipoint tributary operation. "BISYNC" refers to the IBM implementation of bisynchronous protocol. The GENBYTE protocol is a general byte-oriented protocol for use with user-written framing routines, is supported by the DMB32 and the DMF32 device drivers.

The maximum permitted line speed depends on the line interface standard used and must not exceed 19.2 Kb/s for and RS232/V.24 interface. There may be additional line speed restrictions for other line interface standards, refer to the Software Product Descriptions for the synchronous interfaces for more information. Other Digital communications layered products may not necessarily support operation of the synchronous interface at the maximum line speed supported by the device driver. Refer to the appropriate Software Product Description for the Digital communications layered product for precise details of supported configurations.

In all cases, throughput over synchronous lines is dependent on the user application and system environment.

The WAN device drivers all support modem control signal operation. Refer to the Software Product Descriptions for the appropriate synchronous interface for details of the modem signals supported by an individual interface.

Optional Hardware

Additional communications device(s) subject to the limitations described in Table 3 of this SPD are required.

Supported Public Networks

Table 4 describes public PSDNs that are supported with the X.25 software only in the countries shown. In addition to these public PSDNs, certain private PSDNs have been tested by Digital and appropriate profiles are included in the X.25 software.

Since the publication of this SPD, other configurations may have been tested. Consult your local Digital office for any questions concerning other public PSDNs. Digital has designed or adopted the X.25 software to operate with equipment conforming to ISO 8208/7776, with certain private PSDNs that have been approved by Digital, and with public networks in Table 4. Profiles for these private and public PSDNs are included in X.25. Connections to public networks not shown in this table are not supported by Digital.

The presence of a network/country combination in Table 4 shows Digital's commitment to support the X.25 software when using that public X.25 service. It does not necessarily imply that network certification by the particular networking authority has been granted for all or any hardware devices supported by the X.25 software. Contact your local Digital office for up-to-date information regarding supported configurations and certification status.

Digital has designed or adapted the X.25 software to operate with equipment conforming to ISO 7776/8208 with the public networks in the countries listed below and with certain private PSDNs that have been approved by Digital.

Table 4
Supported Public Networks

Country	Public Network and CCITT revision(s)
Argentina	Arpac
Australia	Austpac
Austria	Datex-P
Belgium	DCS
Brazil	Renpac
Canada	Datapac (1976, 1980)
	DPN-NAS
	Faspac
	Infoswitch
Chile	VTRNET
China	CNPAC
Denmark	Datapak
Ireland	Eirpac (1980,1984)
	Postgem
Finland	Datapak
France	Transpac
Germany	Datex-P
Hong Kong	Datapak
	GDN
	INET
	Intelpak
Indonesia	SKDP
Italy	Itapac
Japan	DDX (1976, 1980, 1984)
	ACE Telenet
	CCVAN
	INS-P
	Jaisnet
	Venus-P (1980)
	Venus-LP
	Tymnet
Luxembourg	LUXPAC
Malaysia	Maypac
Mexico	Telepac
Netherlands	Datanet 1
New Zealand	Pacnet
Norway	Datapak
Pakistan	Paknet
Philippines	Datanet
Portugal	Telepac
Singapore	Telepac
South Korea	Dacomnet
Spain	Iberpac
Sweden	Datapak
Switzerland	Telepac (1978, 1980)
Taiwan	Pacnet
Thailand	Samart
	Thaipak
Turkey	Turpac
United Kingdom	PSS
United States	Accunet
	Autonet
	Bell Atlantic
	Compuserve
	ConnNet
	Digipac
	Fedexitc
	Graphnet
	IMPACS
	Infonet
	MarkNet Extended Service
	PPSNET
	Pulselink
	Sinet
	Telenet
	Tymnet
	Western Union PTN-1
	Worldnet

Note: Digital offices in Japan can supply the mandatory additional information that explains the special use of X.25 with the DDX PSDN.

Unless otherwise acknowledged within the product documentation, the trademarks under which the services are offered are proprietary to the respective national PTT, the owning or operating companies, or equivalent organization.

X.25 Software Configuration Guidelines

Table 5 describes the physical hardware configurations supported by X.25 in terms of CPU type and communication interface. It is assumed that there are 128 bytes of user data per packet in an X.25 native mode configuration where data is being sent and received at maximum speeds in both directions. This table indicates the maximum load for certain VAX processors supported at the time of publication.

The numbers given in the table are load costs. Load costs indicate the overhead of full-duplex packet transfer at the indicated speed. The load units are calculated for an application that does no work with transmitted/received data, and indicate a maximum packet rate for that configuration. An idle device incurs negligible load. Systems should be configured to sustain peak load rates.

Table 5

CPUs, Maximum Load and Supported Devices

CPU	Maximum Load in load_units	Supported Device(s)
MicroVAX II	160	DSV11, DPV11-DA
MicroVAX 2000	100	DSH32, DST32
MicroVAX 3100	350	DSH32
MicroVAX 3100 Model 30/40	440	DSW41, DSW42
MicroVAX 3100 Model 80	1000	DSW41, DSW42
MicroVAX 3100 Model 90	2100	DSW41, DSW42
MicroVAX 3300/3400	210	DSV11, DPV11-S
MicroVAX 3500/3600	330	DSV11, DPV11-S
MicroVAX 3800/3900	390	DSV11, DPV11-S
VAXft Model 310	320	DSF32
VAXserver 3100	290	DSH32
VAXserver 6000 Model 210	210	DMB32, DSB32
VAXserver 6000 Model DSB32		
VAXserver 6000 Model 400 Series	300	DMB32, DSB32
VAX–11/730	40	DMF32, DUP11
VAX–11/750	70	DMF32, DUP11
VAX–11/780	120	DMF32, DUP11
VAX–11/785	180	DMF32, DUP11
VAX 8600	400	DMF32, DUP11
VAX 8650	500	DMF32, DUP11
VAX 4000 Model 300	580	DSV11, DPV11-S
VAX 6000 Model 200 Series	250	DMB32, DSB32
VAX 6000 Model 300 Series	350	DMB32, DSB32
VAX 6000 Model 400 Series	450	DMB32, DSB32
VAX 8200/8300	100	DMB32, DSB32
VAX 8250/8350	125	DMB32, DSB32
VAX 8500	360	DMB32, DSB32
VAX 8530	450	DMB32, DSB32
VAX 8550	550	DMB32, DSB32
VAX 8700	700	DMB32, DSB32
VAX 8800/8810/882/ 8830/8840	700	DMB32, DSB32
VAX 8842	700	DMB32, DSB32
VAX 8978	700	DMB32, DSB32

Factors to Consider

Other factors to consider when configuring hardware devices for use with X.25 on a given processor are:

- Hardware configuration limits, such as power supply, backplane space, bus throughput and mapping registers available, and any other restrictions on the number of devices per CPU or per bus must be observed.
- Actual/average packet size in bytes — Generally, doubling the actual packet size will halve the load units consumed for direct memory access (DMA) devices, but increase the load units for character-interrupt devices by approximately 30 percent.
- Table 5 assumes that data is sent and received in each direction of a full-duplex line in equal ratio. If the number of X.25 packets per second is predominantly one-way, the load units per line can be reduced proportionately up to a factor of 50 percent.

The maximum system configuration with a mix of device types is computed by adding the load costs (in load units) for each device up to the limit for a given processor type. Table 6 lists the load units for devices at various speeds.

Table 6

Load Units per Line at Various Speeds

Device Type	Number of Lines	Maximum Speed	Load Units used at Speed (Kb/s)				
			4.8	9.6	19.2	48	64
DMB32	1	64.0	9	18	36	N/S	N/S
DSB32	2	64.0	9	18	36	90	120
	or 1	192.0					
DMF32	1	19.2	9	18	36	N/S	N/S
DPV11-DA*	1	9.6	23	64	N/S	N/S	N/S
DPV11-S							
DST32	1	19.2	10	20	40	N/S	N/S
DSH32	1	19.2	10	20	40	N/S	N/S
DUP11*	1	9.6	23	46	N/S	N/S	N/S
DSV11	2	64.0	9	18	36	90	120
	or 1	256.0					
DSW21	1	256.0	9	18	36	90	120
or DSW41							
DSW42	2	64.0	9	18	36	90	120
	or 1	256.0					
DSF32	2	64.0	9	18	36	90	120

Notes:

- * For more than one character interrupt device, multiply the load units by 50 percent to allow for interaction between devices.
- N/S The use of the device at this speed is not supported by X.25.

Example 1:

A VAX 8500 computer is being used to process X.25 native mode communications. If the system uses one line of a DSB32 at 192 Kb/s and a DMB32 at 19.2 Kb /s, the load unit total is 156. Assuming these devices are fully active, they will consume 156/360 = 43% of the CPU for communications processing. It is inadvisable to add more devices to this CPU, unless it is dedicated to the task of communications processing.

Example 2:

When one line of a DSV11 is fully used on a MicroVAX 3800 at 64 Kb/s, it consumes 120/390 = 31% of the processor in processing X.25 communications.

Certain versions of these products depend on a specific version of the operating system. Refer to the Software Product Descriptions of the product in question to determine which version is required.

SOFTWARE REQUIREMENTS

- OpenVMS Operating System Version 7.0

OpenVMS Tailoring

For Version 7.0 systems, the following OpenVMS classes are required for full functionality of this layered product:

- OpenVMS Required Save Set
- Network Support
- Programming Support

For more information, refer to the OpenVMS VAX Operating System Software Product Description (SPD 25.01.xx). OpenVMS classes and tailoring are discussed in the OpenVMS VAX installation manual.

OPTIONAL SOFTWARE

TCP/IP

A separate TCP/IP protocol stack is required to use the DECnet over IP features in DECnet/OSI. The following TCP/IP products have been tested with DECnet/OSI.

- DEC TCP/IP Services for OpenVMS Version 3.2 or later

  DEC TCP/IP Client Software License: QL-GL7A*-**
  DEC TCP/IP Services Software License: QL-VHRA*-**

  DEC TCP/IP Client Upgrade License: QL-0PHA*-**
  SPD: 25.A4.xx
- Process Software: TCPware
- TGV: Multinet
- The Wollongong Group: PATHWAYS

X.25 Optional License

An X.25 license is required when using X.25 and the LAPB protocol over the WAN. This license is not required when using X.25 and the LLC2 protocol on the LAN.

Software License: QL-071A*-AA
Media and Documentation: Included in DECnet/OSI for OpenVMS VAX media and documentation kits.

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

This product is available as part of the OpenVMS Consolidated Software Distribution on CD-ROM. The software documentation for this product is also available as part of the OpenVMS Online Documentation Library on CD-ROM.

DECnet/OSI for OpenVMS VAX is also available on TK50 streaming tape and 9-track 1600-bit/in magtape for a limited time. Support for H-Kits on this traditional media are being phased out in favor of the CD-ROM distribution.

ORDERING INFORMATION

Software Licenses:	QL-D04A*-AA End System
	QL-D05A*-AA Extended Function
OpenVMS VAX CD-ROM Distribution:	QA-VWJ8A-A8
Software Media:	QA-D04AA-H*
Software Documentation:	QA-D04AA-GZ (Core Documentation Set)
	QA-D04AE-GZ (Programming Supplement)
	QA-D04AF-GZ (X.25 Supplement)
	QA-D04AH-GZ (DECnet/OSI End System User Guide)
Consolidated Distribution Service:	QT-VWJ8A-*8
Software Product Services:	QT-D04A*-**

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

SOFTWARE LICENSING

The DECnet/OSI licenses give users the right to use the software on a single CPU and include the delivery of a License Product Authorization Key (PAK) to enable the DECnet/OSI for OpenVMS software.

To use this software product on additional CPUs, users must purchase a Single-Use License Option for each CPU.

The End System license grants the right to use all the DECnet/OSI features with the exception of the DECdns Server, cluster alias, the OSI application programming interfaces, and the OSI applications gateway.

The Extended Function license grants the right to use the DECnet/OSI end-system features, the DECdns Server, the OSI application gateways, OSI application programming interfaces, and the cluster alias. An Extended Function license is required on at least one node in every cluster configuration to enable the use of the cluster alias.

This software is furnished only under a license. For more information about Digital's licensing terms and policies, contact your local Digital office.

License Management Facility Support

This product supports the OpenVMS VAX License Management Facility.

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

For more information on the License Management Facility, refer to the OpenVMS VAX Operating System Software Product Description (SPD 25.01.xx) or the *License Management Facility* manual of the OpenVMS VAX operating system documentation set.

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

SOFTWARE PRODUCT SERVICES

Prerequisite Support

For the use of X.25 with PSDNs, the customer and Digital must jointly prepare a Network Profile and Customer Support Plan covering all the intended network nodes, their usage of Switched Virtual Circuits (SVCs), Permanent Virtual Circuits (PVCs), and other network facilities, and their support. Without this Network Profile and Customer Support Plan, Digital cannot support the network connections.

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

SOFTWARE WARRANTY

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

APPENDIX A: FILE TRANSFER, ACCESS, AND MANAGEMENT (FTAM)

FTAM software provides communications for performing file operations between open systems. These operations are:

- Copying files between local and remote systems.
- Appending, deleting, or renaming files on open systems.
- Displaying information about files on open systems.

An open system is a computer system that implements the standards for each of the seven layers of the Open Systems Interconnection (OSI) Reference Model for communications as defined by the International Organization for Standardization. An FTAM system is any open system containing an FTAM implementation that conforms to the FTAM standard and includes the implementations of the necessary underlying OSI services.

FTAM implements several standards that define the following components of these layers of the OSI Basic Reference Model: the File Transfer, Access, and Management (FTAM) service element and the Association Control Service Element (ACSE) of the Application layer, the Presentation layer, and the Session layer.

Supported Standards

FTAM conforms to the following OSI standards:

- ISO 8571 — File Transfer, Access and Management service and protocol
- ISO 8650 — ACSE protocol
- ISO 8823 — Presentation protocol
- ISO 8327 — Session protocol

The following table provides a comparison of the supported implementation profiles for different standards bodies and their relationship to each other.

International Standardized Profiles (ISP) ISO 10607	NIST	CEN/CENELEC and EWOS
Part 1: Specification of ACSE, Presentation and Session protocols for use by FTAM	—	—
Part 2: Definition of document types, constraint sets, and syntaxes	—	—
Part 3: AFT11 — Simple File Transfer Service (Unstructured)	T1 — Simple File Transfer	A/111 — iENV 41 204
Part 4: AFT12 (DISP) — Positional File Transfer Service (Flat)[1]	T2 — Positional File Transfer	A/112 — ENV 41 206
Part 5: AFT3 (DISP) — File Management Service	M1 — Management	A/13 — ENV 41 205

[1] AFT12 is not supported by DECnet/OSI

DISP indicates a draft ISP.

FTAM Component Software

The component software includes the user facilities (initiators), responders, management tools, and problem determination tools.

FTAM User Facilities

Users access the FTAM user facilities with the following OpenVMS operating system commands: APPEND, COPY, DIRECTORY, RENAME, and DELETE. These commands operate on files stored on any FTAM system whose implementations are compatible with FTAM. These commands cannot be used for manipulating files on your local system.

Support for Any File Naming Convention

A file designation is system-specific information that identifies a file to its storage system. FTAM software lets users specify files using the naming conventions of the systems where the files reside. FTAM supports the OpenVMS operating system RMS format for file specifications and a comparable style of file-specification format that accommodates non-RMS file designations.

Support for Several File Types

FTAM software can access and transfer files containing both binary and ASCII data. FTAM-1, FTAM-2, FTAM-3, and NBS-9 document types are supported.

FTAM-1 files are unstructured text files, FTAM-2 files are sequential text files, and FTAM-3 files are unstructured binary files. The FTAM-1, FTAM-2, and FTAM-3 document types support the following parameters.

Document Type	String Significance	Universal Class	Maximum String Length
FTAM-1	Not significant	IA5String GeneralString	Presence and absence of parameter
	Fixed	VisibleString GraphicString	Presence of parameter
	Variable	VisibleString GraphicString	Presence and absence of parameter
FTAM-2	Not significant	VisibleString GraphicString	Presence or absence of parameter
FTAM-3	Not significant		Presence or absence of parameter
	Fixed		Presence of parameter

NBS-9 files are NBS file directories.

Flexible and Transparent Access for Local Files

FTAM software treats local files the same way that the OpenVMS operating system file system treats them.

File Transfers

The FTAM COPY command transfers files between compatible FTAM systems without modifying the source file. The facility can transfer files in either direction between the local system and a remote FTAM system. The COPY command can also transfer files between two remote FTAM systems for a local FTAM user.

This command also allows you to append one or more files to a single output file within or between FTAM systems.

FTAM-DAP Gateway

The FTAM-DAP Gateway lets you perform file operations from a non-OSI system that supports the DAP protocol to an OSI system supporting FTAM. Remote users of the gateway do not have to establish accounts on the gateway system to use its capabilities. The gateway does not support the APPEND functionality.

File Deletion

The FTAM DELETE command can delete one or more files on any combination of FTAM systems provided that the user has delete access to those files on the specific FTAM system.

Renaming Requests

The FTAM RENAME command allows you to rename files. The command works on files stored on remote FTAM systems (remote files). The command enables you to change the path name or file name of an existing file. For remote files, you must specify whatever type of information the remote FTAM system requires for specifying files.

Directory Requests

The FTAM DIRECTORY command displays the complete set of FTAM file attributes. Specific options allow users to vary the display of attributes that are meaningful in an OpenVMS operating system environment, for example, date/time of last modification of file name.

FTAM File Error Recovery

FTAM provides file error recovery functionality, both in the COPY initiator command and in the FTAM responder. File error recovery is provided for Class 1, Class 2, and Class 3 type errors as detailed in ISO 8571-4.

Class 1 file error recovery provides only the restart functionality, while Classes 2 and 3 file error recovery provide both the restart and recovery functionality as follows:

- If an internal error is detected in the data transfer regime, Class 1 recovery restarts the data transfer regime by retransmitting the file data beginning at the negotiated checkpoint within the data transfer regime.
- Class 2 error recovery provides for the re-establishment of the select and open regimes, and also allows for the retransmission of file data beginning at a negotiated checkpoint within the data transfer regime.
- Class 3 error recovery provides full recovery by re-establishing a lost FTAM association and its select and open regimes. Class 3 recovery will then restart the data transfer regime by retransmitting the file data beginning at the negotiated checkpoint within the data transfer regime.

All restart and recovery operations and procedures are completely transparent to the user.

Management and Problem Determination Tools

FTAM software supplies a number of management tools, including an installation verification procedure (IVP), a tracing utility, event logging, and informational and error messages.

FTAM Installation Verification Procedure (IVP)

The FTAM IVP sets up outbound and inbound application associations. A connection is made to your local system (as a loopback test). The FTAM IVP checks that your installation is able to set up and release presentation and session connections. It tests the FTAM software by starting a responder and reading the attributes of a file with the DIRECTORY command.

FTAM Tracing Utility

The FTAM tracing utility (OSITRACE) is a tool for identifying problems in protocol exchanges between your local system and any remote FTAM system. The tracing utility captures protocol exchanges and transcribes them into easily read text; OSITRACE data is written to SYS$OUTPUT.

The FTAM tracing utility monitors data exchanges for individual associations. The tracing utility can trace data originating from the following components: FTAM (DATA, PROTOCOL, and STRUCTURING), ACSE, Presentation, and Session.

OSI Address Lookup using X.500

The FTAM software is capable of retrieving network addresses from the X.500 directory. You may use this functionality in conjunction with or instead of retrieving addresses from a local repository.

FTAM Event Logging

For event logging, the FTAM responder writes records to OSIF$RESPONDER.LOG.

Requirements for Compatibility with FTAM

FTAM lets an open system perform a specific set of file transfer, access and management activities with any open system having a compatible FTAM implementation.

The Protocol Implementation Conformance Statement (PICS) provides more information about Digital's FTAM implementation.

APPENDIX B: VIRTUAL TERMINAL

DECnet/OSI Virtual Terminal is Digital Equipment Corporation's implementation of the ISO Virtual Terminal Basic Class standard, which is comprised of the service definition (ISO 9040) and the protocol (ISO 9041). The DECnet/OSI Virtual Terminal software adheres to these standards, thereby providing interactive access between DECnet/OSI systems and other multivendor terminal systems and host systems that also adhere to the ISO Virtual Terminal Basic Class standard.

Virtual Terminal is implemented as an Application Service Element (ASE) of the OSI Application layer.

DECnet/OSI Virtual Terminal can run over Transport Layer Classes 0, 2 or 4 over CONS, and TP4 over CLNS. DECnet/OSI Virtual Terminal can also run over TCP/IP networks using RFC 1006.

Virtual Terminal provides terminal/initiator (for a local user) and host/responder (for the remote user) capabilities. Terminal/responder and host/initiator are not supported.

Supported Standards

Virtual Terminal conforms to the following OSI standards:

- ISO 9041 — Virtual Terminal Protocol - Basic Class
- ISO 8650 — ACSE protocol
- ISO 8823 — Presentation protocol
- ISO 8327 — Session protocol

Virtual Terminal Features

Virtual Terminal supports the following features:

- Class of Service
  - Basic class (character cell terminals)
- Mode of Operation
  - Asynchronous Mode (A-Mode)
- Profile Support
  - Default A-mode (per ISO 9040)
  - A-mode Generalized Telnet (adheres to OIW Stable Agreements)
  - A-mode Transparent (adheres to OIW Stable Agreements)
  - A-mode Telnet 1988 (adheres to OIW Stable Agreements)
- Functional Units
  - destructiveBreak
  - structuredCOs
  - urgentData

- Supported Gateways
  - Bidirectional VT/Telnet
  - Bidirectional VT/LAT
- On-Line Help

OSI Address Lookup using X.500

The Virtual Terminal software is capable of retrieving network addresses from the X.500 directory. This functionality may be used in conjunction with or instead of retrieving addresses from a local repository.

Command Mode

Command mode allows the user to execute commands that can modify the characteristics of the Virtual Terminal association with the remote application.

Trace Utility

The Virtual Terminal tracing utility (OSITRACE) is a tool for identifying problems in protocol exchanges between your local system and any remote system. The utility captures protocol exchanges and transcribes them into easily read text.

The tracing utility monitors data exchanges for individual associations. The utility can trace data originating from the VT, ACSE, Presentation, and Session components.

APPENDIX C: OSI APPLICATION PROGRAMMING INTERFACES

FTAM API

The FTAM API supports the OSI standard ISO 8571—Information Processing Systems—Open Systems Interconnection—File Transfer, Access, and Management.

The FTAM API provides the interface to the FTAM protocol machine provided in DECnet/OSI. Communications between the application code and the FTAM protocol machine are carried out using ASN.1 encoded data structures. The API supports the following functions:

- Abort
- Begin-Group
- Cancel
- Change-Attributes
- Create
- Close
- Data
- Data-End
- Delete
- Deselect
- End-Group
- Initialize
- Open
- Read-Attributes
- Read
- Select
- Terminate
- Transfer-End
- Write

ACSE API

The Association Control Service Entity (ACSE) API supports the following OSI standards

- ISO 8650—Information Processing Systems—Open Systems Interconnection—Protocol Specification for the Association Control Service Element.
- ISO 8649—Information Processing Systems—Open Systems Interconnection—Service Definition for the Association Control Service Element.

The ACSE API provides the interface to the Associate and Release services. The services provided are

- Associate
- Release
- Abort
- Redirect

The Redirect service is not an ACSE service. It allows applications to redirect an incoming association to another process on the local system.

Presentation Layer API

The implementation of the Presentation layer in the toolkit supports the following OSI standards:

- ISO 8823—Information Processing Systems—Open Systems Interconnection—Connection-Oriented Presentation Protocol Specification
- ISO 8822—Information Processing Systems—Open Systems Interconnection—Connection-Oriented Presentation Service Definition
- ISO 8327—Information Processing Systems—Open Systems Interconnection—Basic Connection-Oriented Session Protocol Specification
- ISO 8326—Information Processing Systems—Open Systems Interconnection—Basic Connection-Oriented Session Service Definition.

The Presentation API provides the interface to the Presentation layer services and, by passthrough, to the equivalent Session layer services. The API supports the following services:

- Alter-Context
- Data
- Capability-Data
- Expedited-Data
- Typed-Data
- Token-Please
- Token-Give
- Control-Give
- Sync-Major
- Sync-Minor
- Resynchronize
- Exception-Report
- Activity-Start
- Activity-Interrupt
- Activity-Resume
- Activity-Discard
- Activity-End

Session Layer API

The implementation of the Session layer supports the following OSI standards:

- ISO 8327—Information Processing Systems—Open Systems Interconnection—Basic Connection Oriented Session Protocol Specification
- ISO 8326—Information Processing Systems—Open Systems Interconnection—Basic Connection Oriented Session Service Definition.

The Session layer supports ISO Session version 1 and version 2. Session version 1 allows up to 512 octets of user data on a service. Session version 2 supports the restrictions imposed by the National Institute of Standards and Technology allowing up to 10,240 octets of data on a service.

The Session API provides the interface to the Connect and Release services as follows:

- Connect
- Release
- Abort
- Redirect

The Redirect service is not a Session service. It allows applications to redirect an incoming connection to another process on the local system.

The Session API also provides the following services

- Data
- Capability-Data
- Expedited-Data
- Typed-Data
- Token-Please
- Token-Give
- Control-Give
- Sync-Major
- Sync-Minor
- Resynchronize
- Exception-Report
- Activity-Start
- Activity-Interrupt
- Activity-Resume
- Activity-Discard
- Activity-End

ROSE API

The Remote Operations Service Element (ROSE) supports the following OSI standard:

- ISO 9072 — Information Processing Systems — Text Communication — Remote Operations

ROSE supports interactive applications in a distributed open systems environment. It is a service for multivendor distributed processing.

The ROSE functionality provides a mechanism that allows for the encoding and decoding of the Remote Operations Protocol Control Information, as defined in ISO 9072, for the following services:

- Invoke
- Result
- Error
- Reject