Software Product Description

PRODUCT NAME: X.25 Network-Level Developer's Toolkit™ for UNIX®, Version R1.10 **SPD 38.97.00**

DESCRIPTION

The X.25 Network-Level Developer's Toolkit for UNIX is a Digital Distributed Software product from Eicon Technology Corporation. It allows programmers to write end-user applications that can communicate through the OSI PC Gateway™ for UNIX (described in Software Product Description (SPD) 38.94.xx).

The X.25 Network-Level Developer's Toolkit supports PVCs and SVCs including incoming only, outgoing only, and two-way connections. It can control up to 254 virtual circuits.

The toolkit supports optional user facilities such as NUI (Network User Identification), reverse charging, Call User Data, CUGs (Closed User Groups), throughput class negotiation, and window size negotiation.

The X.25 Network-Level Developer's Toolkit also offers support for X.25 M-bit, Q-bit, D-bit, interrupt packets, and reset packets.

Features

The X.25 Network-Level Developer's Toolkit includes the following features:

- Full support for the CCITT recommendation X.25 (1980 and 1984) including Extended Format and ISO 8208
- Supports up to 254 virtual circuits
- Window size configurable from 1 to 127
- Packet size 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096 bytes
- Supports permanent virtual circuits and three types of switched virtual circuits: incoming only, outgoing only, and two-way
- Support for X.25 M-bit, Q-bit, and D-bit
- Support for interrupt and reset packets

The toolkit provides the following user facilities:

- Flow control parameter negotiation including packet size and window size
- Throughput class negotiation
- Closed User Group (CUG) selection
- CUG with outgoing Access selection
- Bilateral CUG selection
- Reverse charging
- Fast select
- Network User Identification (NUI)

Table 1
Toolkit Functions

x25accept( ) Generate a CALL ACCEPTED packet from the application

x25alloc( ) Allocate buffer space to send and receive data using the *x25send( )* and *x25recv( )*

x25call( ) Set up a virtual circuit to a specified DTE, allowing limited choice of parameters

x25cancel( ) Cancel a pending request

x25listen( ) Accept an incoming call

x25mkconn( ) Dynamically obtain a connection identifier for an already existing connection

x25packetstats( ) Return packet-level statistics for a specified connection

x25pcause( ) Display an X.25 library clearing cause message

Table 1 (Cont.)
Toolkit Functions

x25cause( ) Return an appropriate X.25 networks library clearing cause number	**x25pdiag( )** Display an X.25 library diagnostic message
x25causemsg( ) Return an appropriate X.25 library clearing cause message	**x25perror( )** Display an X.25 library error message
x25delconn( ) Delete a connection identifier without completing a hangup	**x25pvccall( )** Set up a virtual circuit to a specified DTE via a PVC
x25diag( ) Return an appropriate X.25 library diagnostic number	**x25recv( )** Receive data from a remote DTE
x25diagmsg( ) Return an appropriate X.25 library diagnostic message	**x25reset( )** Reset a virtual circuit
x25done( ) Get request status	**x25resetconfirm( )** Acknowledge a reset indication
x25error( ) Return an appropriate X.25 library error number	**x25send( )** Send data to a remote DTE
x25errormsg( ) Return an appropriate X.25 library error message	**x25sendconfirm( )** Acknowledge a packet received with the D-bit
x25exit( ) Clear all connections used by the application program	**x25sendexp( )** Send expedited data (interrupt data) to the remote DTE
x25flowctl( ) Enable/Disable flow control for incoming data at the networks level	**x25sendexpconfirm( )** Send an expedited data confirmation packet
x25free( ) Free buffer space allocated by the *x25alloc( )* function	**x25status( )** Get the status of a session
x25getconn( ) Get the logical session number and port number of a connection	**x25version( )** Return the X.25 Network-Level Developer's Toolkit version number
x25hangup( ) Terminate a connection and any associated pending request	**x25xcall( )** Set up a virtual circuit to a specified DTE, allowing extended use of parameters
x25hangupconfirm( ) Generate the CLEAR CONFIRMATION packet in the basic format	**x25xhangup( )** Terminate a connection using the extended packet format
x25init( ) Initialize the X.25 library functions	**x25xhangupconfirm( )** Confirm the reception of clear indication using extended packet format
x25linkstats( ) Return link-level statistics for a specified connection	**x25xlisten( )** Accept an incoming call and establish a virtual circuit, allowing extended use of parameters

HARDWARE REQUIREMENTS

Processors Supported

80386- and 80486-based Digital personal computers

SOFTWARE REQUIREMENTS

- SCO™ UNIX Release 3.2 V4 (see SPD 39.81.xx)
- OSI PC Gateway for UNIX Version V2R1 (see SPD 38.94.xx)

The use of a "C" or a Pascal compiler is required (where applicable) to develop application programs.

GROWTH CONSIDERATIONS

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

ORDERING INFORMATION

Software License, Documentation, and Media (5.25-inch and 3.5-inch floppy disks): QB-MJZAA-WW

SOFTWARE LICENSING

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

SOFTWARE PRODUCT SERVICES

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

SOFTWARE WARRANTY

NONE

THIS SOFTWARE PRODUCT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY KIND EITHER EXPRESS OR IMPLIED.

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