# TCPware for OpenVMS

TCPware for OpenVMS is the complete networking solution for Hewlett-Packard Corporation's VAX, Alpha and Integrity systems. TCPware for OpenVMS turns VAX, Alpha and Integrity computers into powerful application servers in multi-platform environments. It integrates OpenVMS systems with virtually any other system through industry-standard TCP/IP.

## Fast and Efficient

TCPware for OpenVMS takes full advantage of the distinct architecture of OpenVMS for VAX, Alpha, and Itanium systems. TCPware implements lower-layer protocols (TCP, UDP, and IP) as device drivers, focusing on minimal CPU loading. This provides peak performance so that TCPware integrates cleanly into the OpenVMS environment.

With support extended now to OpenVMS versions 5.5-2 through 8.4, TCPware supports the OpenVMS Communications Interface (VCI), a high-speed interface to Ethernet, FDDI, and Token Ring LAN drivers.

## Easy to Install and Operate

Installing TCPware is easy using the usual VMSINSTAL installation procedure and, if you configure all services and utilities, typically takes 30 minutes or less. A menu-driven configuration option is available also.

You can control most components in TCPware by means of a single Network Control Utility (NETCU) that simplifies network management and allows you to manage TCPware security. Using NETCU, you can:

- Start and stop network interfaces
- Configure network hosts dynamically
- Add and remove services
- Provide secondary addresses for cluster failover
- Display and modify routing tables
- Display network counters and connections
- Enable gateway and multi-casting support

TCPware lets you start and stop each of its components without rebooting the entire system and affecting other products.

## Configuration Support

TCPware for OpenVMS supports VAX, Alpha and Itanium servers running various versions of OpenVMS. When each node in a VMS cluster shares a common system disk, the cluster needs to store just one copy of most TCPware files. You require only a few system-specific configuration files on each machine that runs the software.

TCPware supports Symmetric Multi-Processing (SMP) for OpenVMS. Also supported by TCPware are Class A, B, C, and D (multi-cast) networks. For subnetted networks, Classless Inter-Domain Routing (CIDR) variable subnet masks are on bit boundaries.

## The Core Features of TCPware…

- Provides fast and efficient operation that is designed and optimized for VAX, Alpha and Integrity systems
- Installs and operates easily, with no connectivity limitations
- Includes security features for Secure Shell (SSH) V1 and V2, Secure Copy Protocol v2 (SCP2), Secure File Transfer Protocol (SFTP), access restriction, advanced packet filtering, Intrusion Prevention System (IPS), Kerberos protocol, IP Security Options, token authentication, and FTP over TLS.
- Supports most system and hardware configurations, including LAN devices, VAX WAN Device Drivers, Fast Ethernet (with OpenVMS v7.0), Serial Line IP, Point-to-Point, ATM, and X.25
- Follows OpenVMS standards closely for command syntax, basic security, and compatibility with standards products, such as TCP/IP Services for OpenVMS
- Provides access to a wealth of utilities and services, including:
    - Domain Name Services (DNS) BIND 9.6.1 (including DDNS)
    - Berkeley "R" Commands (RLOGIN, RSH, RMT, RCD) and Services, (rlogin, shell, and rmt)
    - Line Printer Services (LPR commands and LPD Server)
    - Internet Printing Protocol (IPP)
    - Terminal Server Print Services
    - SNMP Services, including SNMP Multiplexing (SMUX), and Agent Protocol (AgentX)
    - DECwindows Transport Interface and XDM support
    - Network Time Protocol (NTP) version 4.2
    - TIMED, the Time Synchronization Protocol (TSP) daemon
    - Network Control Utility (NETCU)
    - Master Server Process (NETCP)
    - DECnet over IP
    - PING, TCPDUMP, NSLOOKUP, TALK, and other utilities
    - FTP and SMTP accounting statistics
    - NFS Version 3 Server and NFS Version 2 Client
- Allows additional third-party support via compatibility with HP and other products, services, and utilities
- Provides services to maximize network efficiency:
    - Dynamic Host Configuration Protocol (DHCP) Server and Safe-failover
    - Dynamic Host Configuration Protocol (DHCP) Client
    - Cluster alias failover
    - Multiple gateways, routing, and multi-casting
    - Dynamic TCP/IP load balancing
    - Paired Network Interface Support
- Provides programming support, including a Socket Library, QIO interfaces, FTP and TELNET programming libraries, ONC RPC services, and TCP/IP Services for OpenVMS compatibility

### HP Secure Web Server (Apache)

TCPware for OpenVMS v5.9 supports HP's secure Web server.

## Multiple Interfaces (Paired Network) on a Common Ethernet Cable

TCPware for OpenVMS supports systems that have multiple interfaces on a common Ethernet, FDDI, ATM, or Token Ring cable. TCPware internally links the interfaces together. If an interface fails, a linked interface can be used. If data is to be transmit-ted on an interface that happens to be busy, TCPware assigns the data to the least busy linked interface for transmission.

## Dynamic Host Configuration (DHCP) Server v3.0

TCPware provides a Dynamic Host Configuration Protocol (DHCP) server that assigns network addresses to hosts based on a local reusable pool. DHCP also supports groups of clients on remote subnets on your network via relay agents. With these features, you can configure local host addresses quickly without relying on outside sources. DHCP supports Dynamic DNS (DDNS; see RFC 2136).

DHCP also includes safe-failover support, which allows for two servers (primary and secondary) to share a configuration and to service clients using the same address pool. The safe-failover protocol guards against duplication of address assignments during network failures, even if the network is partitioned so the primary and secondary servers cannot communicate and are independently leasing addresses.

## Dynamic Host Configuration (DHCP) Client

The DHCP client resides on the client host and dynamically sets the network configuration. The TCPware DHCP client communicates with a DHCP server to get an IP address and other configuration information. It uses this information to configure the network parameters of the host and to start up the network.

## LAN Devices

TCPware for OpenVMS operates with standard Digital Ethernet/802.3, FDDI, and Token Ring network controllers. DECnet, Local Area Transport (LAT), and Local Area VMScluster (LAVC) software can share these controllers concurrently with TCPware. There is also support for ATM controllers by means of the OpenVMS 7.1 (and later) Classical IP over ATM and LAN emulation support.

## IP-over-X.25

TCPware for OpenVMS supports sending IP datagrams over certain X.25 packet switching networks using HP's VAX Packetnet System Interface (PSI) product. You can connect separate TCP/IP LANs over packet switching data networks (PSDNs) or other X.25 WANs.

## Serial Line IP (SLIP)

You can send IP datagrams over serial lines using any standard OpenVMS serial line as a SLIP device. Dialup, dedicated serial lines, and Compressed SLIP (CSLIP) are supported.

## Point-to-Point Protocol

TCPware for OpenVMS supports the Point-to-Point (PPP) interface for sending IP datagrams over serial links. PPP provides more enhanced features than the SLIP interface, such as error detection and automatic negotiation of header compression, and supports PAP and CHAP.

## IP-over-DECnet

You can send IP datagrams and connect separate TCP/IP LANs over DECnet WAN links.

## VAX WAN Device Drivers

TCPware for OpenVMS supports the DSV11, DSB32, and DST32 synchronous interfaces for point-to-point links between sys-tems over a digital WAN. These cards support the DDCMP, LAPB, LAPBE, and HDLC protocols at speeds up to 256 Kbps.

## PATHWORKS (Advanced Server), DECnet/OSI, and NTDS Support

TCPware for OpenVMS provides TCP/IP support for PATHWORKS (Advanced Server) v5.0 and later, DECnet/OSI v0 and later through its PWIPDRIVER, and NTDS.

## Third-Party Application Support

The TCPware for OpenVMS emulation of standard OpenVMS facilities supports several software products developed for compatibility with OpenVMS. For a complete list of companies and their products, refer to the Process Software website (www.process.com).

## Network Performance

TCPware for OpenVMS includes services that provide fast and efficient network operation and that minimize downtime.

### Classless Inter-Domain Routing

TCPware for OpenVMS includes support for Classless Inter-Domain Routing (CIDR). CIDR eliminates address class distinctions, relies on address masks that fall on bit boundaries, and aggregates routing information to reduce exponential growth in routing tables.

### Gateway Routing Daemon

TCPware for OpenVMS includes the Gateway Routing Daemon (GateD) that consolidates RIP, DCN Hello, OSPF, EGP, BGP, and the Router Discovery Protocol into one distributed routing service. GateD supports route and protocol masks and uses the subnets supported with CIDR. GateD includes a rich language that is flexible in meeting any routing need.

### Routing and Multiple Gateway Support

TCPware for OpenVMS includes routing and gateway capabilities for WANs and complex LANs, and supports multiple default gateways essential for automatic failover. TCPware detects gate-ways that are possibly down and rotates addresses in the routing table so that a gateway is always available. This minimizes sending datagrams to out-of-service gateways and maintains network stability.

### Dynamic TCP/IP Load Balancing

The Domain Name Services supports dynamic TCP/IP Load Balancing, primarily for TCP-based applications such as TELNET. This allows the least-loaded systems running TCPware in a TCP/IP cluster to appear first in response to DNS host name requests. A TCP/IP cluster can include independent systems, hosts anywhere, and several OpenVMS clusters, provided they have TCP/IP connectivity.

### Cluster Alias Failover

Cluster Alias Failover lets one node in a cluster take over incoming connection requests from a client system if the servicing node goes down.

Cluster Alias Failover is primarily for UDP applications, such as NFS. However, you can also use Cluster Alias Failover with TCP applications, such as FTP and TELNET, to establish a connection to the server.

# Network Services Support

## Berkeley R Commands and Services

TCPware for OpenVMS incorporates the Berkeley remote access commands ("R" commands). These are UNIX client and server facilities for remote access to hosts in a TCP/IP network. They include the RLOGIN remote login command and the RSH remote execution command.

Local users can back up their files on remote (UNIX system) magnetic tapes using the RMT client. Remote users can back up their files on local magnetic tapes using the RMT server.

The Berkeley R commands use standard OpenVMS security facilities plus "host equivalence" files. For added security, you can use full Kerberos authentication with RLOGIN and RSH. You can also use Secure Shell (SSH), using TCPware's security features.

TCPware for OpenVMS also supports RCD, which provides local users the ability to access remote CD-ROM drives as if they were local drives.

## Path MTU Discovery

Support for Path MTU Discovery improves performance when large packets of data are sent over TCP. Path MTU Discovery causes TCP to segment data into the largest datagrams that can be transmitted to the remote host without fragmentation along the path.

## DECwindows Transport Interface

TCPware for OpenVMS contains a DECwindows transport interface that operates over TCP/IP. This lets you run DECwindows applications on remote workstations running TCP/IP, and X Window System applications on local Alpha, VAX and Itanium workstations.

## X Display Manager Server

TCPware for OpenVMS provides an X Display Manager (XDM) server to manage remote X terminals. When an X display starts, it communicates with the XDM server through the UDP-based X Display Manager Control Protocol (XDMCP). The XDM server creates a DECwindows login process, which then prompts remote X display users to login and create a DECwindows session.

## Domain Name Services

TCPware provides the Domain Name Services (DNS) that implement the Berkeley Internet Name Domain (BIND) server standard. You can configure DNS for a client or server. DNS includes Dynamic DNS (DDNS), updates, DNS notify support, and enhanced control. With DNS notify support, the primary server notifies the secondary servers when zone changes occur, and the secondary server can then immediately initiate zone transfers rather than wait for the polling interval to expire.

## Terminal Server Print Services

The Terminal Server Print Services allow system managers to configure the print queues using standard OpenVMS printer operations, including the autostart feature. Users have access to IP terminal server-based printers plus printers that connect directly to Ethernet as they would any other OpenVMS printer.

## Line Printer Services

TCPware for OpenVMS implements the client and server ends of the BSD 4.3 Line Printer Protocol for various print devices connected to LPD servers and connected directly to the network.

Using LPS, you can:

- Print local files on remote printers.
- Remove print jobs from remote queues.
- Display job status in remote print queues.

LPS supports the UNIX commands lpr, lpq, and lprm, as well as the PRINT command. LPS includes the LPD server.

## Internet Printing Protocol (IPP)

TCPware for OpenVMS supports version 1.0 of the Internet Printing Protocol. The IPP print symbiont is an OpenVMS print symbiont working with the OpenVMS printing subsystem to implement an IPP Client. It allows printing over a network to printers and servers that support the IPP v1.0 network printing protocol. The TCPware IPP print symbiont provides standard commands for advanced printer functionality, such as double-sided printing. The `TCPWARE IPP SHOW` utility allows a user to learn the capabilities supported by an IPP server. This utility queries the server and displays the supported attributes. The program can be used to check on the capabilities of a given server.

## SNMP Services

SNMP Services implements the agent (server) end of the Simple Network Management Protocol (SNMP). The agent supports management objects defined in the SNMP Management Information Base (MIB II), plus subagents serving private MIBs using an API.

## SNMP Multiplexing (SMUX)

The SNMP Multiplexing (SMUX) protocol is an SNMP subagent extension protocol. Each subagent or peer registers a MIB subtree with the SNMP Agent. Requests for objects residing in a registered MIB subtree are passed from the SNMP Agent using the SMUX protocol to the subagent. The subagent passes the result of an SNMP query back to the SNMP agent.

## SNMP Agent X

Agent X is a standardized protocol allowing the list of managed objects available from an SNMP agent to be dynamically extended. By using Agent X directly, writers of TCP/IP services can allow the state of the service to be queried and controlled remotely. This can be useful if the service does not have a user interface, or runs under batch, or as a detached process. The HP Insight Manager uses the SNMP extensibility provided by Agent X to allow remote examination and notification of system conditions that may need attention. Insight Manager is available on AXP systems with OpenVMS 7.1 and higher, and all Itanium systems.

## Network Time Synchronization Facilities

TCPware for OpenVMS supports two types of time synchronization between network hosts:

- Network Time Protocol version 4.2 (includes NTPQ and NTPDC)
- TIMED, the Time Synchronization Protocol (TSP) daemon

## Master Server Process

The master server process invokes all server processes, which are present when a connection is active. The system manager can easily add and remove servers any time by entering NETCU commands.

The master server also:

- Logs all activity for security monitoring
- Can invoke user-written server processes

- Can restrict access to services based on the source Internet address

### DECnet over IP

The DECnet over IP service permits two machines running DECnet to communicate using IP links. This is an important service for TCP/IP WANs that might link several local sites running DECnet with others that run only TCP/IP.

### Multi-casting

TCPware for OpenVMS supports full Class D IP multi-casting (level 2) to host groups. Multi-casting support is available for the UDPDRIVER, IPDRIVER, BGDRIVER, INETDRIVER, and Socket Library programming interfaces.

## Programming Support

### Socket Library

TCPware for OpenVMS provides a socket library of C routines (also accessible from other high-level languages) to facilitate application development. These routines support the UNIX socket functions for raw, stream, and datagram sockets. Socket library calls include socket and lookup operations, and byte order and Internet address conversion functions.

### QIO Programming Interface

TCPware for OpenVMS provides a QIO interface for application programmers to develop their own networking programs using the TCP, IP, and UDP protocols. The QIO interface includes operations used to open and close connections or ports, and to transfer data over a connection or port. All high-level languages can use this interface.

### Compatibility with TCP/IP Services for OpenVMS

TCPware for OpenVMS is compatible with TCP/IP Services for OpenVMS, allowing applications written for products, such as DECwindows, PATHWORKS (Advanced Server), and DECmcc, to run transparently on top of TCPware. The interface is the BGDRIVER.

### INETDRIVER Services

TCPware for OpenVMS provides the INETDRIVER Services that support the Stanford Research Institute (SRI) QIO interface. This provides a one-to-one mapping between the UNIX socket functions and the OpenVMS $QIO system services.

### IP External Interface

TCPware for OpenVMS provides a programmable IP interface that is easy to use and adds full TCP/IP networking for nearly any network controller.

### ONC RPC Programming Services

ONC RPC Services is a software development tool based on the Open Network Computing (ONC) protocols for version 4 of remote procedure calls (RPC). TCPware supports two sets of ONC RPC Services: one for the HP C Socket Library and one for the TCPware Socket Library. ONC RPC Services include:

- A shareable runtime library
- RPCGEN compiler
- RPCINFO utility
- TCP and UDP synchronous transports

- Broadcast RPC and batch RPC
- RTL and XDR routines

## Enhanced Security Features

The security features in TCPware for OpenVMS provide data protection and security over the network that far exceeds what normal networks offer. This added security is important with the ever-increasing number of LANs, WANs, and hosts on the network. Network security prevents unauthorized use of systems, services, and network information.

TCPware offers seven types of security services:

- Secure Shell (SSH) v1 and v2
- Secure File Transfer (SCP and SFTP)
- FTP over TLS (FTPS)
- Outgoing access restrictions
- Packet filtering
- Kerberos password authentication
- IP security option (IPSO)
- SSH Publickey Assistant
- CMPCLIENT
- CERTVIEW
- Intrusion Prevention System (IPS)

### Secure Shell (SSH) v1 Client and Server

TCPware SSH (Secure Shell) v1 is a program for logging into and executing commands on a remote system. It replaces rlogin, rshell, TELNET programs, and provides secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can be forwarded over the secure channel. SSH connects and logs into the specified hostname.

The TCPware SSH v1 implementation is based on the version 1.3.7 protocol. The Secure Shell daemon (SSHD) is the daemon program for SSH v1 that listens for connections from clients. When the SSHD daemon starts, it generates a server RSA key (normally 768 bits). This key is regenerated every hour (the time may be changed in the configuration file) if it has been used, and is never stored on disk. A new daemon is created for each incoming connection. The multiple encryption algorithms supported by SSH v1 are IDEA (the default), DES, 3DES, BLOWFISH, and ARCFOUR.

A client program (SSH) is provided with TCPware, but any SSH client that uses SSH v1 protocol may be used to access the server. Examples of such programs are FISSH and TCPware SSH on OpenVMS systems; TTSSH, SecureCRT, F-Secure SSH Client, and PuTTY on Windows-based systems; and other SSH programs on UNIX-based systems.

SSH v1 offers the following server system authentications: rhosts, rhosts-rsa, rsa challenge-response, and password.

SSH v1 and v2 offer break-in and intrusion detection, session termination, X11 forwarding, and port forwarding.

### Secure Shell (SSH) v2

TCPware's SSH v2 implementation is based on the V2 protocol and the WRQ RSIT 6.1.0.16 code base. While SSH v2 is generally regarded to be more secure than SSH v1, both protocols are offered by TCPware. Although the protocols are

incompatible, they may exist simultaneously on a TCPware system. The TCPware server front-end identifies what protocol a client desires to use, and will create an appropriate server instance for that client.

The SSH2 server and client are compiled from unaltered cryptographic source which is FIPS 140-2 Level 2 compliant.

The client and server together, using the Diffie-Hellman key-exchange method, determine a 256-bit random number to use as the "session key". This key is used to encrypt all further communications in the session.

The multiple encryption algorithms supported by SSH v2 are 3DES (the default), TWOFISH, BLOWFISH, DES, CAST-128, and ARCFOUR.

SSH v2 offers the following server system authentications: host-based, public-key, and password.

SSH can be used to create a secure tunnel between two systems. It is possible to have one end of this tunnel point to an FTP server and provide a secure channel for FTP transfers. Some SSH servers and clients recognize the FTP PORT and PASV commands and replies and can provide protection for the data channel as well. To use this method an SSH connection must be established between the two systems before the FTP connection is established, which adds inconvenience or uses resources even when there are no transfers being done. With this method SSH provides data privacy and integrity, server identification verification and privacy for the user password. FTP provides any data format conversion that is necessary between the two systems.

### Publickey Assistant

The publickey assistant can be used to add, remove, and list public keys that are stored on a remote server.

### CMPCLIENT

Allows users to enroll certificates by connecting to a CA (certification authority) and using the CMPv2 protocol for enrolling a certificate. The user may supply an existing private key when creating the certification request or allow a new key to be generated.

### CERTVIEW

Allows users to view and validate certificates, and, optionally, to output the information from a certificate that is formatted correctly to use when creating the SSH certificate mapping configuration.

### CERTTOOL

The CERTTOOL utility is used for different needs concerning X.509 certificates in PKCS#10 and PKCS#12 format. The CERTVIEW tool can be used for certificate viewing and validation.

For PKCS#10, CERTTOOL creates certificate requests, allowing the user to specify specific keyUsage and extended-KeyUsage flags.

For PKCS12, CERTTOOL creates a PKCS#12 package containing any number of private keys and certificates. The final PFX package is encoded with a HMAC and by default contains one password protected safe, which contains all the other objects in an unshrouded format.

### Secure Copy Protocol v2 (SCPv2)

SCP2 is an evolving file transfer protocol, and not all implementations will offer all levels of functionality. The basic functionality is binary file transfers. TCPware supports BINARY and ASCII transfers with SCP2, and will also transfer VMS file characteristics when the remote system has the capability. When operating with systems that do not support the full range of transfer mechanisms that TCPware offers, TCPware uses various methods to improve the chances that files will be useful upon transfer.

TCPware uses the defined extensions in the protocol to transfer information about the OpenVMS file header characteristics such that when a file is transferred between two OpenVMS systems running TCPware v5.7 or later; or MultiNet v5.1 or later, the file header information will also be transferred and the file will have the same format on the destination system as it had on the source system. Also, when a file is transferred to a non-OpenVMS system, a method has been provided to translate those files that can be translated into a format that will be usable on the remote system. Files that are transferred from non-OpenVMS systems are stored as stream files on the OpenVMS system, which provides compatibility for text files from those systems.

## Secure File Transfer Protocol v2 (SFTP2)

SFTP2 is an FTP-like client that can be used to transfer files over a network. SFTP2 transfers the files through ssh2 connections to ensure that the file transport is secure. In order to connect using SFTP2, you need to make sure that sshd2 is running on the remote host that you are connecting to.

SFTP2 is an evolving file transfer protocol, and not all implementations will offer all levels of functionality. The basic functionality is binary file transfers. TCPware supports BINARY and ASCII transfers with SFTP2, and will also transfer VMS file characteristics when the remote system has the capability. When operating with systems that do not support the full range of transfer mechanisms that TCPware offers, TCPware uses various methods to improve the chances that files will be useful upon transfer.

## FTP over TLS (FTPS)

FTPS allows users to establish a secure, encrypted connection to the FTP server for user authentication. File transfers can also be secured at the user's option. FTPS offers better performance than SFTP as only a single process is used for encrypting and transferring the data. FTPS provides more reliable interchange of files between dissimilar systems as it uses the well-developed FTP protocol.

## Outgoing Access Restrictions

Outgoing access restrictions screen the remote TCP applications to which a local user can gain access. The system manager can implement these restrictions using NETCU commands.

## Packet Filtering

Packet filtering restricts the datagrams a network interface can receive. You can filter datagrams by protocol (IP, ICMP, UDP, or TCP), source and destination address, or source destination port (UDP and TCP). The system manager can implement these restrictions using NETCU commands.

## Intrusion Prevention System (IPS)

Components of TCPware, including SSH, FTP, SNMP, SMTP, TELNET, IMAP and POP3 have been instrumented to report various failures (“events”) such as invalid login attempts, etc., to a central filter server.

The filter server correlates reported events via rulesets and may implement a packet filter on an interface based on the results of the event correlation. This can be based on either the source address, essentially blocking all traffic of a particular protocol (e.g., IP, UDP, etc.) from a system; or on the destination address and port, blocking traffic only to that port.

Rules may be implemented such that certain source networks or addresses are excluded from event correlation, or have event correlation applied with different parameters, allowing the same rule to be applied differently, for example, to internal versus external network traffic.

An API is supplied so that TCPware users may incorporate this event reporting into their own applications, as well as implementing the corresponding rulesets for event correlation for their applications in the filter server.

### Kerberos Authentication

TCPware for OpenVMS provides Kerberos v4 authentication. Kerberos, an established authentication protocol, relies on a secure server to ensure login security. Kerberos uses data encryption to produce password ciphertext on TCP/IP networks.

With Kerberos, hosts prove their identity to other systems without transmitting "cleartext," or human-readable passwords. Their systems do not have to rely on the network for security.

### Ephemeral Port Obfuscation

Ephemeral ports are allocated at a random offset from the previous one to improve security.

### Kerberos Applications

The following TCPware for OpenVMS applications allow Kerberos authentication for added security:

- RCP command
- RLOGIN command and rlogin services
- RSH command and rsh service
- TELNET-OpenVMS client and server

To requesting hosts, the Kerberos server issues tickets that contain keys to lock or unlock encrypted tickets, which in turn contain keys to lock or unlock encrypted passwords. The server is available to any heterogeneous Kerberos clients and servers from different vendors running different operating systems.

The Kerberos server includes a Key Distribution Center (KDC) and the Kerberos Administration (KADM) functions also.

### Kerberos Administration Server

The Kerberos Administration Server provides an administration model so that system managers have remote access to the Kerberos database and remote users can change their passwords.

### Kerberos User

Using the Kerberos User (KUSER) model, users can obtain and manage Kerberos tickets to use with their secured applications.

### IP Security Option

TCPware for OpenVMS implements the IP Security Option (IPSO), a protocol developed for the United States Department of Defense to label datagrams with defined classification levels and established government protection authorities. Systems can screen which of these labeled datagrams to receive or transmit to ensure confidentiality of incoming and outgoing data.

## Additional Features

TCPware for OpenVMS provides the TALK Utility and the TCPDUMP utility.

The TALK utility enables remote users to share terminal messages in split windows in real time.

The TCPDUMP utility is a useful mechanism for tracking TCP packets by printing information contained in the packet headers.

# FTP-OpenVMS

**FTP-OpenVMS** provides TCP/IP File Transfer Protocol networking services for OpenVMS computer users that need to transfer files from one computer's system to another. The number of simultaneous connections to FTP-OpenVMS is limited only by the available system resources.

FTP supports RFC 4217 - Securing FTP with TLS, which allows the user to log in over an encrypted connection and for data to be transferred over an encrypted connection.

## FTP-OpenVMS Features...

- Provides a Client and a Server
- Handles both UNIX and VMS command interface types in interactive mode, as a single-line command, or in command procedures
- Maintains consistent file protection and security
- Supports OpenVMS file types, DECnet access, and remote DCL commands
- Offers a number of additional features to enhance ease of use in all modes and functions
- Centralized logging
- Records accounting and statistical information from enabled services
- Supports RFC 4217 - Securing FTP with TLS
- FTP supports STREAM mode transfer RESTARTS

### Client and Server Support

FTP-OpenVMS supports a File Transfer Protocol client and server. You can transfer files in both directions between local and remote systems that implement the TCP/IP and FTP protocols.

### OpenVMS and UNIX Commands

Using the command line interface, you can initiate file transfers using native OpenVMS commands or equivalent UNIX-style commands interactively or with command procedures.

### Session Accounting and Statistics

TCPware can record accounting information from services that have been enabled. Currently this includes FTP and SMTP. The accounting information includes information about when a network session took place and how much data was transferred.

### Full File Protection and Security

FTP-OpenVMS uses maximum OpenVMS file protection for each user. You can limit access for ANONYMOUS users or CAPTIVE accounts. Network managers can log all attempted connections to a local host. FTP-OpenVMS supports token authentication and full OpenVMS break-in detection and evasion.

### FTP over SSL

FTP has been enhanced with support for RFC 4217 - Securing FTP with TLS so that user authorization and file transfers can now take place over secured connections.

### FTP Subroutine Library

FTP-OpenVMS contains an FTP layer callable subroutine library. These subroutines allow user-written programs to establish FTP connections with and transfer FTP session data between other systems supporting an FTP Server.

### Ease of Use

FTP-OpenVMS provides the same environment to remote users as if they were logged in locally and supports many features to make file transfers easy:

- Multi-line recall of up to 20 lines
- Startup command files

- Automatic file transfer format determination
- Record structure transfer support
- STRU O VMS and VMS PLUS support
- Multi-homed hosts support (if Client-FTP needs to reach a host that has multiple internet addresses, it tries all possible addresses)
- Centralized logging
- Records accounting information from enabled services

# TELNET-OpenVMS

**TELNET-OpenVMS** provides complete virtual terminal networking services to OpenVMS systems by implementing the TELNET and TCP/IP protocols. TELNET-OpenVMS users have immediate access to any remote system (such as UNIX, Linux, and Windows) that supports TCP/IP and TELNET, eliminating the need for dedicated terminals and serial ports.

## TELNET-OpenVMS Features...

- Provides a Client and a Server
- Represents a fast, efficient design
- Permanence of NTA devices
- Supports Kerberos v4 authentication
- Provides a familiar interface to UNIX and OpenVMS users
- Provides a customizable TN3270 mode
- Provides a library of callable functions

### Client and Server Support

TELNET-OpenVMS provides a TELNET client and server. Users on a TCPware system can login to remote systems, and users on remote systems can login to a TCPware system via TELNET-OpenVMS.

### Designed for Efficiency

Server-TELNET is for high-bandwidth applications. TCPware implements the Server as an OpenVMS device driver, operating with minimal CPU overhead.

Server-TELNET performs processing within a port driver for the TTDRIVER class driver. This makes the server a standard OpenVMS terminal device that is fully compatible with all TTDRIVER QIOs.

### Permanence of NTA Devices

TELNET-OpenVMS provides the option to permanently assign NTA devices, making NTA setup and operations similar to LAT outgoing connections.

### Full Password Protection (Kerberos)

TELNET-OpenVMS fully supports username and password protection by using the optional Kerberos v4 authentication scheme, provided with the token authentication security feature.

### OpenVMS and UNIX Commands

You can use native OpenVMS commands or a UNIX-style command interface.

### TN3270 Mode

Client-TELNET supports TN3270 mode, providing IBM 3270-class terminal emulation for local OpenVMS terminals. Remote IBM hosts must support TELNET Servers.

Client-TELNET maps the OpenVMS keyboard to emulate IBM 3270 keyboard functions. You can use the default keyboard mappings or customize them.

### TN3270 Internationalization

Client-TELNET supports the conversion of Western European EBCDIC character sets to corresponding OpenVMS character sets for TN3270 mode.

## TELNET Subroutine Library

TELNET-OpenVMS contains a TELNET layer callable subroutine library. These subroutines allow user-written programs to establish TELNET connections with and transfer TELNET session data to other systems supporting TCP/IP and TELNET. Programmers can use the subroutine library to:

- Open and close a TELNET connection.
- Allocate and deallocate a TELNET connection control block (CCB).
- Get and set the value of a CCB field.
- Send and receive data.
- Send TELNET commands.
- Abort a TELNET connection.

## TELNET Protocol Options

TELNET-OpenVMS supports the TELNET protocol options BINARY, ECHO, END-OF-RECORD, SUPPRESS-GO-AHEAD, TERMINAL-TYPE, and TRANSMIT-BINARY.

## Additional Features

TELNET-OpenVMS also offers:

- Multi-line recall of up to 20 command lines
- Definable keys
- Startup command files
- OpenVMS process spawning
- Control character mapping
- Interactive, online help
- Support for multi-homed hosts; if Client-TELNET needs to reach a host that has multiple internet addresses, it tries all possible addresses
- A multi-session client that supports up to ten simultaneous normal mode sessions
- Support for X Display Location option to set the user's current X display location on the remote end
- Support for the Remote Flow Control option for disabling and enabling flow control

# NFS-OpenVMS Client

**NFS-OpenVMS Client** implements the NFS client side of the Network File System (NFS) v2 and v3 protocols, providing access to file systems on remote NFS servers. Authorized users on the local Alpha, VAX or Itanium systems have transparent and multi-threaded access to remote NFS servers, such as UNIX/Linux or Windows servers.

## Filesystem Mount Flexibility

Users can obtain access to remote filesystems by mounting them. The client provides flexibility so you can mount any level of the NFS Server Filesystem directory structure onto any level of the Client Filesystem directory structure, subject to OpenVMS Record management Services (RMS) restrictions.

## Complete File Protection

The client fully supports system, directory, and file protection. Access confirmation to NFS files is through user ID mappings in a PROXY database and group ownership mappings in a GROUP database. You can quickly load each of these databases to implement changes without remounting the file disk. The client supports Network Lock Manager as well as the standard file locking and sharing protocols.

## File Format

The client adheres to NFS file organization and record format specifications so that you can write files back to the server.

The client preserves file structures across the network, and maintains file attributes the NFS protocol does not address by using attributes data files (ADFs). Automatic format handling treats existing UNIX files as sequential, variable-length, carriage-return-carriage-control files on your OpenVMS system.

## NFS-OpenVMS Client features...

- Supports flexible filesystem mounting in all forms, including:
  - Background mounting
  - Occluded mounting
  - Automounting
  - Shared mounting
  - Overmounting
  - Configuration file mounting
  - Setting access parameters
  - Displaying mounts
- Maintains consistent file protection and security
- Supports OpenVMS file types using Attribute Data Files to preserve file type information
- Maps filename characters to pre-serve file naming conventions be-tween systems
- Provides command interface at the DCL level or the NETCU utility, or through command procedures
- Provides dynamic PROXY reload-ing

## Filename Mapping

Even though OpenVMS uses different conventions for naming files from those on an NFS server, special characters are not rejected. Instead, the client maps file name characters between the operating systems. Users in each environment can continue to use the naming conventions to which they are accustomed, subject to the RMS restrictions on file name length.

## Flexible Command Interface

You can mount filesystems and display mount information either interactively at the DCL or NETCU level, or by using command procedures.

The command syntax, shown next, is convenient and straightforward:

```
NFSMOUNT server “path” [mount [logical]]
```

An example command is:

```
NFSMOUNT LILAC “/usr/users” NFS1:
```

# NFS-OpenVMS Server

**NFS-OpenVMS server** implements the server side of the NFS v2 and v3 protocol, providing access to filesystems on your OpenVMS host to remote client NFS users. These remote users can run a variety of operating systems, including UNIX/Linux and Windows. The NFS server lets your network share data among different systems. This minimizes hardware costs by eliminating data duplication. The server supports NFS over UDP and TCP, and can also export files to TCPware NFS-OpenVMS client systems.

## Network File Systems v3 (NFS)

TCPware supports the NFS v3 server (RFC 1813), which provides increased performance over the NFS v2 server due to protocol changes which allow NFS servers to return results of file attributes in response to normal operations; and return file handles and attributes during directory read requests which eliminates subsequent lookup operations; separate calls are no longer required.

## File Operations

The NFS server supports all normal file operations, even those on multi-volume disks. NFS clients can use the server system's files as if they were local files. The server supports the MOUNT and Port Mapper protocols and operations. It also supports symbolic links and hard links.

System resources are the only limitations to the number of simultaneous users. A multi-threaded architecture provides fast, high-performance service for many clients, while keeping processor overhead to a minimum.

## Complete File Protection

The server fully supports stem, directory, and file protection. Access to OpenVMS files is restricted to preapproved clients named in an EXPORT database. C- and ACL-based protection using an easily reloadable PROXY database that maps between NFS UID/OID and OpenVMS user accounts. The server uses the OpenVMS UIC and user access rights to validate all file access. The server even enforces OpenVMS disk quotas.

To further increase security, the network administrator can assign "rights identifiers" to NFS users, restrict remote mounts to superusers only, and track attempted access violations.

## File Format

The server allows clients to read OpenVMS files in their most commonly used formats, including sequential, variable-length, and variable with fixed-length control (VFC), without having to manually convert these files. You can use OpenVMS disks for information sharing as well as file storage.

## Filename Mapping

Even though OpenVMS uses different conventions for naming files from those on an NFS client system, special characters are not rejected. The server maps file name characters between the operating systems. Users in each

## NFS-OpenVMS Server features...

- Supports file operations in all forms, including:
  - Create or remove directory
  - Create, remove, or rename file
  - Get or set attributes
  - Get filesystem statistics
  - Look up file or read directory
  - Read from or write to file
  - Maintains consistent file protection and security
- Maps filename characters to pre-serve file naming conventions among systems
- Supports standard protocols for locking and file sharing across dif-ferent systems
- Provides information and tools for tuning your server system to max-imum performance
- Provides dynamic PROXY reload-ing
- Provides XQP+ multi-threading support for improved file lookup, creation, and access
- Routing features
- Supports OpenVMS ODS2 and ODS5 file systems.

environment can continue to use the naming conventions to which they are accustomed, subject to the RMS restrictions on file name length.

## Standard Protocols for File Sharing

NFS-OpenVMS Server supports these protocols for file sharing:

- **UNIX Support Protocols:** The server supports the Network Lock Manager and Status Monitor RPC protocols. These provide advisory UNIX System V locking and PC file sharing. This lets you coordinate access to file and file records using standard methods in a distributed environment.
- **PC Support Protocols:** The server supports the PCNFSD protocol, providing PC users with access to OpenVMS filesystems and the ability to use OpenVMS print queues.
- **Performance Tuning:** The server generates statistics and optionally logs security violations, MOUNT requests, errors, and other activities to help you tune the performance of the NFS server system. Tuning parameters control such things as datagram sizes, cache sizes, and the number of server threads.

# SMTP-OpenVMS

**SMTP-OpenVMS** provides complete mail transfer networking services by implementing the TCP/IP and Simple Mail Transfer Protocol (SMTP) networking standards for OpenVMS systems. You can implement mail rejection rules, necessary for blocking mail relaying and adding anti-spamming capabilities to TCPware. You can also deliver files as base64-encoded MIME messages by way of VMSmail.

## SMTP features...

- Provides full SMTP Client and Server
- Provides IMAP4 Server
- Provides POP3 Server
- Uses standard MX records when using DNS
- Supports ARPA-standard formats and addresses, and mail request expansion
- Automatically stores, forward, and relays mail traffic as needed
- Supports performance tuning parameters
- Provides additional functionality, such as gateway to other mail services (such as ALL-IN-1)
- Spam prevention
- Records accounting information from enabled services (See FTP for details)

### SMTP Client and Server Support

SMTP-OpenVMS provides an SMTP client and server. Users on a system running SMTP-OpenVMS can send mail messages to and receive mail messages from users on systems that support SMTP and TCP/IP.

### IMAP4 Server

The Internet Message Access Protocol (IMAP) server lets the mail program of your IMAP-compliant client access remote mes-sage storage as if the storage were local. TCPware's implementation is based on IMAP version 4, revision 1.

IMAP4 and the Post Office Protocol (POP3), described in the next section, operate differently. IMAP4 retains the message on the server, while POP3 retrieves the message and stores it offline on the client, thus deleting it from the mail server. IMAP4 allows you to access your mail from more than one client workstation simultaneously.

### POP3 Server

The Post Office Protocol version 3 (POP3) multi-threaded server provides a way for users on remote hosts (such as PCs) who do not want to maintain their own message transport systems to retrieve mail from an OpenVMS mail server's incoming mailbox.

### Transparent User Interface

Users have a transparent interface to the SMTP messaging system from within the OpenVMS MAIL utility. All features of OpenVMS MAIL message processing are available, including:

- All OpenVMS MAIL commands, including SET FORWARD
- Alias names, mailing lists, and special mail headers
- Distribution name lists
- Automatic notification of incoming mail
- Reading incoming mail using OpenVMS MAIL
- Carbon copy (CC:) recipients

### Store, Forward, and Relay

SMTP-OpenVMS notifies users automatically of incoming or undeliverable mail, defers mail delivery to unavailable hosts, and can forward mail to a central mail handling machine. You can choose to forward all mail or only mail with unknown addresses to the central mail handling machine.

## ARPA Standard Message Formats

SMTP-OpenVMS supports standard message formats and addresses used in the ARPA Internet community.

SMTP-OpenVMS user names have the format:

```
SMTP%"address[,address[,...]]"
```

Network mailbox addresses have the basic format:

```
username[@domain]
```

The domain is the name of the destination host, according to DNS standards.

## Mail Exchanger (MX) Records

SMTP-OpenVMS uses mail exchanger (MX) records on systems using DNS. MX records specify which hosts can accept mail for a domain. If the first attempt to deliver mail fails, SMTP-OpenVMS tries each MX record until it finds a host that can accept the mail.

## Performance Tuning

You can set parameters at runtime to customize and enhance SMTP-OpenVMS performance. These parameters include:

- Connection timeout value
- Delivery check and retry intervals
- Maximum message life

## Additional Features

SMTP-OpenVMS also provides the following features:

- Any user can be the postmaster
- SMTP-OpenVMS can function as a gateway between SMTP and DECnet and foreign mail products

## Standards and RFCs

TCPware conforms to the following military and Internet Engineering Task Force (IETF) standards.

| Military Standard | Mil-Std |
|---|---|
| Internet Protocol | 1777 |
| Transmission Control Protocol | 1778 |
| File Transfer Protocol | 1780 |
| Simple Mail Transfer Protocol | 1781 |
| TELNET Protocol and Options | 1782 |

| IETF RFC Title | RFC # |
|---|---|
| User Datagram Protocol (STD 6) | 768 |
| DARPA Internet Protocol Specification | 791 |
| Internet Control Message Protocol | 792 |
| Transmission Control Protocol | 793 |
| Simple Mail Transfer Protocol (STD 10) | 821 |
| Standard for the Format of Internet Text Messages (STD 11) | 822 |
| An Ethernet Address Resolution Protocol | 826 |
| TELNET Protocol Specification (STD 8) | 854 |
| TELNET Option Specification (STD 8) | 855 |
| TELNET Binary Transmission (STD 27) | 856 |
| TELNET Echo Option (STD 28) | 857 |
| TELNET Suppress Go Ahead Option (STD 29) | 858 |
| Echo Protocol (STD 20) | 862 |
| Discard Protocol (STD 21) | 863 |
| Character Generator Protocol (STD 22) | 864 |
| Quote of the Day Protocol | 865 |
| Daytime Protocol (STD 25) | 867 |
| Time Protocol (STD 26) | 868 |
| TELNET End of Record Option | 885 |
| Trailer Encapsulations | 893 |
| Transmission of IP Datagrams over Ethernet Networks | 894 |
| Reverse Address Resolution Protocol | 903 |
| Broadcasting Internet Datagrams (STD 5) | 919 |
| Broadcasting Datagrams in the Presence of Subnets (STD 5) | 922 |
| Internet Standard Subnetting Procedure (STD 5) | 950 |
| Bootstrap Protocol (BOOTP) | 951 |
| File Transfer Protocol (STD 9) | 959 |
| Mail Routing and the Domain System (STD 14) | 974 |
| XDR: External Data Representation Standard | 1014 |
| Domain Administrators Guide | 1032 |
| Domain Administrators Operations Guide | 1033 |
| Domain Names—Concepts and Facilities | 1034 |
| Standard for IP Datagrams over IEEE 802 Networks | 1042 |
| Network Systems HYPERchannel Protocol Specification | 1044 |
| Transmission of IP Datagrams over Serial Lines: SLIP | 1055 |
| RPC: Remote Procedure Call Protocol, version 2 | 1057 |
| TELNET Window Size Option | 1073 |
| TELNET Terminal Speed Option | 1079 |

| | |
|---|---|
| TELNET Terminal-Type Option | 1091 |
| NFS: Network File System Protocol Specification | 1094 |
| TELNET X Display Location Option | 1096 |
| DNS Encoding of Network Names and Other Types | 1101 |
| U.S. Department of Defense Security Options for Internet Protocol | 1108 |
| Host Extensions for IP Multi-casting (level 2) (STD 5) | 1112 |
| Compressing TCP/IP Headers for Low-Speed Serial Links | 1144 |
| Management Information for TCP/IP Internets (STD 17) | 1155 |
| A Simple Network Management Protocol (SNMP) (STD 15) | 1157 |
| Line Printer Daemon Protocol | 1179 |
| New DNS RR Definitions | 1183 |
| Path MTU Discovery | 1191 |
| MIB-II | 1213 |
| SNMP MUX Protocol and MIB | 1227 |
| Tunneling IPX Traffic through IP Networks | 1234 |
| BSD rlogin | 1282 |
| Finger User Information Protocol | 1288 |
| Network Time Protocol (version 3) | 1305 |
| TCP Extension for High Performance Options | 1323 |
| DNS NSAP RRs | 1348 |
| Type of Service in the Internet Protocol Suite | 1349 |
| The TFTP Protocol (Revision 2) (STD 33) | 1350 |
| Multi-protocol Interconnect on X.25/ISDN in Packet Mode | 1356 |
| TELNET Remote Flow Control Option | 1372 |
| Transmission of IP and ARP over FDDI Networks (STD 36) | 1390 |
| IP Multi-cast over Token-Ring Local Area Networks | 1469 |
| Encoding Header Field for Internet Messages | 1505 |
| CIDR Applicability Statement | 1517 |
| CIDR Address Allocation Architecture | 1518 |
| CIDR Address Strategy | 1519 |
| Dynamic Host Configuration Protocol | 1541 |
| Classical IP and ARP over ATM | 1577 |
| The Point-to-Point Protocol (PPP) (STD 51) | 1661 |
| Assigned Numbers (STD 2) | 1700 |
| TFTP Blocksize Option | 1783 |
| NFS version 3 Specification | 1813 |
| Post Office Protocol—version 3 (STD 53) | 1939 |
| Internet Message Access Protocol—version 4 rev 1 | 2060 |
| Domain Names—Implementation and Specification | 2065 |
| Dynamic Host Configuration Protocol | 2131 |
| DHCP Options and BOOTPD Vendor Extensions | 2132 |
| Domain Names—Implementation and Specification | 2136 |
| Secure Domain Name System (DNS) Dynamic Update | 2137 |
| Internet Printing Protocol/1.0: Encoding and Transport | 2565 |
| Internet Printing Protocol/1.0: Model and Semantics | 2566 |
| Design Goals for an Internet Printing Protocol | 2567 |
| Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol | 2568 |
| Mapping between LPD and IPP Protocols | 2569 |
| Internet Printing Protocol/1.0: Implementer's Guide | 2639 |

| | |
|---|---|
| SNMPAgent Extensibility (AgentX) Protocol version 1 | 2741 |
| Definitions of Managed Object for Extensible SNMP Agents | 2742 |
| Network Services Monitoring MIB | 2788 |
| Mail Monitoring MIB | 2789 |
| Extensions to FTP | 3659 |
| The Secure Shell (SSH) Protocol Architecture | 4251 |
| The Secure Shell (SSH) Connection Protocol | 4252 |
| The Secure Shell (SSH) Transport Layer Protocol | 4253 |
| The Secure Shell (SSH) Connection Protocol | 4254 |

# Services, Documentation, and Ordering Information

## Technical Services

Process Software's Technical Services Program has a well-deserved reputation for excellence. Services include consulting, train-ing, software maintenance, support, online resources, and 24-hour support - in short, everything you need to keep your Process Software products and your network operating at peak efficiency.

## Consulting

A comprehensive suite of programs is available on a host of topics, including TCPware installation and configuration, DNS setup and use, network security, troubleshooting, and others.

## Hot Line Support

Networking experts are available by telephone, e-mail, or fax. Optional 24-hour support is also available.

## Updates

All maintenance customers with current service contracts receive automatic software and documentation updates of major releases.

## Training

A wide range of educational services can be provided at your site, at regional training locations throughout North America, or at our own training facility in Framingham, MA.

## Documentation

Comprehensive documentation for all TCPware products includes user guides, installation and configuration information, management functions and utilities, programming facilities, and network security. Documentation in HTML and PDF format is included on your product CD, and is available in HTML format on Process Software's web site, www.process.com.

You can find Frequently Asked Questions (FAQs) on the Tech Support web page on the Process Software web site.

## Ordering Information

TCPware for OpenVMS is shipped on CD-ROM and is available for download via FTP.

## Hardware and Software Requirements

TCPware for OpenVMS requires one or more of the following hardware devices:

- HP Ethernet controller
- HP FDDI controller
- HP Token Ring controller (except DEQRA)
- IP-over-x.25 controller
- HP controller for VAX WAN device drivers
- Network Systems HYPERchannel controller (for VAX)
- Proteon proNET-10/80 controller (for VAX)
- Classical IP-over-ATM

TCPware for OpenVMS requires, at a minimum, these operating system versions:

- VAX/VMS 5.5-2 and later
- OpenVMS Alpha 6.2 and later
- OpenVMS IA64 8.2 and later
- OpenVMS IA64 8.2 and later

## About Process Software

Process Software is a premier supplier of infrastructure software solutions to mission critical environments. We deliver customer-centric and innovative IP-based technologies to our customers worldwide, and provide them with superior customer support and service.

Process Software
959 Concord Street
Framingham, Massachusetts 01701-4682

Telephone:
U.S./Canada 1- (800) 722-7770
International 1- (508) 879-6994

FAX: 1- (508) 879-0042

Web: http://www.process.com

E-mail: info@process.com