TCPware® for OpenVMS User's Guide

Part Number: N-6004-60-NN-A

January 2014

This manual describes how to use the network services provided by the TCPware for OpenVMS product.

Revision/Update: This is a revised manual.

Operating System/Version: VAX/VMS V5.5-2 or later, OpenVMS VAX V6.0 or later, OpenVMS Alpha V6.1 or later, or OpenVMS I64 V8.2 or later

Software Version: 6.0

Process Software
Framingham, Massachusetts
USA

The material in this document is for informational purposes only and is subject to change without notice. It should not be construed as a commitment by Process Software. Process Software assumes no responsibility for any errors that may appear in this document.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The following third-party software may be included with your product and will be subject to the software license agreement.

judgement as to where in this file the bootstrap license should appear.
4. The acronym "ISC", when used in this license or generally in the context of works covered by this license, is an abbreviation for the words "Internet Software Consortium."
5. A distribution, as referred to hereafter, is any file, collection of printed text, CD ROM, boxed set, or other collection, physical or electronic, which can be distributed as a single object and which contains one or more works covered by this license.
6. You may make distributions containing covered files and provide copies of such distributions to whomever you choose, with or without charge, as long as you obey the other terms of this license. Except as stated in (9), you may include as many or as few covered files as you choose in such distributions.
7. When making copies of covered works to distribute to others, you must not remove or alter the Bootstrap License. You may not place your own copyright message, license, or similar statements in the file prior to the original copyright message or anywhere within the Bootstrap License. Object files and executable files are exempt from the restrictions specified in this clause.
8. If the version of a covered source file as you received it, when compiled, would normally produce executable code that would print a copyright message followed by a message referring to an ISC web page or other ISC documentation, you may not modify the file in such a way that, when compiled, it no longer produces executable code to print such a message.
9. Any source file covered by this license will specify within the Bootstrap License the name of the ISC distribution from which it came, as well as a list of associated documentation files. The associated documentation for a binary file is the same as the associated documentation for the source file or files from which it was derived. Associated documentation files contain human-readable documentation which the ISC intends to accompany any distribution.

If you produce a distribution, then for every covered file in that distribution, you must include all of the associated documentation files for that file. You need only include one copy of each such documentation file in such distributions.

Absence of required documentation files from a distribution you receive or absence of the list of documentation files from a source file covered by this license does not excuse you from this from this requirement. If the distribution you receive does not contain these files, you must obtain them from the ISC and include them in any redistribution of any work covered by this license. For information on how to obtain required documentation not included with your distribution, see: http://www.isc.org.

If the list of documentation files was removed from your copy of a covered work, you must obtain such a list from the ISC. The web page at http://www.isc.org contains pointers to lists of files for each ISC distribution covered by this license.

It is permissible in a source or binary distribution containing covered works to include reformatted versions of the documentation files. It is also permissible to add to or modify the documentation files, as long as the formatting is similar in legibility, readability, font, and font size to other documentation in the derived product, as long as any sections labeled CONTRIBUTIONS in these files are unchanged except with respect to formatting, as long as the order in which the CONTRIBUTIONS section appears in these files is not changed, and as long as the manual page which describes how to contribute to the Internet Software Consortium (hereafter referred to as the Contributions Manual Page) is unchanged except with respect to formatting.

Documentation that has been translated into another natural language may be included in place of or in addition to the required documentation, so long as the CONTRIBUTIONS section and the Contributions Manual Page are either left in their original language or translated into the new language with such care and diligence as is required to preserve the original meaning.

10. You must include this license with any distribution that you make, in such a way that it is clearly associated with such covered works as are present in that distribution. In any electronic distribution, the license must be in a file called "ISC-LICENSE".

If you make a distribution that contains works from more than one ISC distribution, you may either include a copy of the ISC-LICENSE file that accompanied each such ISC distribution in such a way that works covered by each license are all clearly grouped with that license, or you may include the single copy of the ISC-LICENSE that has the highest version number of all the ISC-LICENSE files included with such distributions, in which case all covered works will be covered by that single license file. The version number of a license appears at the top of the file containing the text of that license, or if in printed form, at the top of the first page of that license.

11. If the list of associated documentation is in a seperated file, you must include that file with any distribution you make, in such a way that the relationship between that file and the files that refer to it is clear. It is not permissible to merge such files in the event that you make a distribution including files from more than one ISC distribution, unless all the Bootstrap Licenses refer to files for their lists of associated documentation, and those references all list the same filename.

12. If a distribution that includes covered works includes a mechanism for automatically installing covered works, following that installation process must not cause the person following that process to violate this license, knowingly or unknowingly. In the event that the producer of a distribution containing covered files accidentally or wilfully violates this clause, persons other than the producer of such a distribution shall not be held liable for such violations, but are not otherwise excused from any requirement of this license.

13. COVERED WORKS ARE PROVIDED "AS IS". ISC DISCLAIMS ALL WARRANTIES WITH REGARD TO COVERED WORKS INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

14. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OF COVERED WORKS.

Use of covered works under different terms is prohibited unless you have first obtained a license from ISC granting use pursuant to different terms. Such terms may be negotiated by contacting ISC as follows:

Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
Tel: 1-888-868-1001 (toll free in U.S.)
Tel: 1-650-779-7091
Fax: 1-650-779-7055
Email: info@isc.org
Email: licensing@isc.org

DNSSAFE LICENSE TERMS
This BIND software includes the DNSsafe software from RSA Data Security, Inc., which is copyrighted software that can only be distributed under the terms of this license agreement.

The DNSsafe software cannot be used or distributed separately from the BIND software. You only have the right to use it or distribute it as a bundled, integrated product.

The DNSsafe software can ONLY be used to provide authentication for resource records in the Domain Name System, as specified in RFC 2065 and successors. You cannot modify the BIND software to use the
DNSsafe software for other purposes, or to make its cryptographic functions available to end-users for other uses.

If you modify the DNSsafe software itself, you cannot modify its documented API, and you must grant RSA Data Security the right to use, modify, and distribute your modifications, including the right to use
any patents or other intellectual property that your modifications depend upon.

You must not remove, alter, or destroy any of RSA's copyright notices or license information. When distributing the software to the Federal Government, it must be licensed to them as "commercial computer software" protected under 48 CFR 12.212 of the FAR, or 48 CFR 227.7202.1 of the DFARS.

You must not violate United States export control laws by distributing the DNSsafe software or information about it, when such distribution is prohibited by law.

THE DNSSAFE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER. RSA HAS NO OBLIGATION TO SUPPORT, CORRECT, UPDATE OR MAINTAIN THE RSA SOFTWARE. RSA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO ANY MATTER WHATSOEVER, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.
If you desire to use DNSsafe in ways that these terms do not permit, please contact:
RSA Data Security, Inc.
100 Marine Parkway
Redwood City, California 94065, USA
to discuss alternate licensing arrangements.

Secure Shell (SSH). Copyright © 2000. This License agreement, including the Exhibits (“Agreement”), effective as of the latter date of execution (“Effective Date”), is hereby made by and between Data Fellows, Inc., a California corporation, having principal offices at 675 N. First Street, 8th floor, San Jose, CA 95112170 (“Data Fellows”) and Process Software, Inc., a Massachusetts corporation, having a place of business at 959 Concord Street, Framingham, MA 01701 (“OEM”).

Portions copyright 1988 - 1994 Epilogue Technology Corporation.

Copyright (c) 1998-2007 The OpenSSL Project. All rights reserved.

If the examples of URLs, domain names, internet addresses, and web sites we use in this documentation reflect any that actually exist, it is not intentional and should not to be considered an endorsement, approval, or recommendation of the actual site, or any products or services located at any such site by Process Software. Any resemblance or duplication is strictly coincidental.

Contents

Preface

Introducing This Guide

This guide describes the TCPware products, components, and features, and the user environment and functions. It is an introduction for all users, as well as a procedural guide for end users.

What You Need to Know Beforehand

Before using TCPware, you should be familiar with:

- Computer networks in general
- HP's OpenVMS operating system and file system

How This Guide Is Organized

This guide has the following contents:

- Part I, *Introduction*—Introduces and provides a functional overview of the TCPware for OpenVMS products, components, and features.
- Part II, *User Functions*—Provides user instructions for the following TCPware components and features, arranged in chapters alphabetically:
  - FTP-OpenVMS
  - Kerberos authentication user commands
  - Network print functions (Line Printer Services and Terminal Server Print Services)
  - Remote Compact Disk (RCD) and Remote Magnetic Tape (RMT)
  - Remote Copy Program (RCP)
  - RLOGIN
  - Remote Shell (RSH)
  - Simple Mail Transfer Protocol (SMTP)
  - TALK
  - TELNET-OpenVMS
  - Trivial File Transfer Protocol (TFTP)
  - Token Authentication User Functions
  - WHOIS
  - Secure Shell (SSH)
- Appendixes, including a list of references and a glossary of terms.

Online Help

You can use help at the DCL prompt to find the following:

- Topical help—Access TCPware help topics only as follows:

```
$ HELP TCPWARE [topic]
```

The topic entry is optional. You can also enter topics and subtopics at the following prompt and its subprompts:

TCPWARE Subtopic?

Online help is also available from within certain TCPware components: FTP-OpenVMS Client and Server, Network Control Utility (NETCU), TELNET-OpenVMS Client, NSLOOKUP, and TRACEROUTE. Use the HELP command from within each component.

Example: `NETCU>HELP [topic]`

- Error messages help—Access help for TCPware error messages only as follows:

```
$ HELP TCPWARE MESSAGES
```

If the error message is included in the MESSAGES help, it identifies the TCPware component and provides a meaning and user action. See the `Instructions` under `MESSAGES`.

Obtaining Customer Support

You can use the following customer support services for information and help about TCPware and other Process Software products if you subscribe to our Product Support Services. (If you bought TCPware products through an authorized TCPware reseller, contact your reseller for technical support.) Contact Technical Support directly using the following methods:

- Electronic Mail

  E-mail relays your question to us quickly and allows us to respond, as soon as we have information for you. Send e-mail to support@process.com. Be sure to include your:

  – Name
  – Telephone number
  – Company name
  – Process Software product name and version number
  – Operating system name and version number

  Describe the problem in as much detail as possible. You should receive an immediate automated response telling you that your call was logged.

- Telephone

  If calling within the continental United States or Canada, call Process Software Technical Support toll-free at 1-800-394-8700. If calling from outside the continental United States or Canada, dial 1-508-628-5074. Please be ready to provide your name, company name, and telephone number.

- World Wide Web

  There is a variety of useful technical information available on our World Wide Web home page, http://www.process.com (select Customer Support).

- Internet Newsgroup

  You can also access the VMSnet newsgroup, vmsnet.networks.tcp-ip.tcpware.

Licensing Information

TCPware for OpenVMS includes a software license that entitles you to install and use it on one machine. Please read and understand the Software License Agreement before installing the product. If you want to use TCPware on more than one machine, you need to purchase additional licenses. Contact Process Software or your distributor for details.

Maintenance Services

Process Software offers a variety of software maintenance and support services. Contact us or your distributor for details about these services.

Reader's Comments Page

TCPware guides may include Reader's Comments as their last page. If you find an error in this guide or have any other comments about it, please let us know. Return a completed copy of the Reader's Comments page, or send e-mail to techpubs@process.com.

Please make your comments specific, including page references whenever possible. We would appreciate your comments about our documentation.

Documentation Set

The documentation set for TCPware for OpenVMS consists of the following:

- ***Release Notes*** for the current version of TCPware for OpenVMS-—For all users, system managers, and application programmers. The Release Notes are available online on your TCPware for OpenVMS media and are accessible before or after software installation.
- ***Installation & Configuration Guide***—For system managers and those installing the software. The guide provides installation and configuration instructions for the TCPware for OpenVMS products.
- ***User's Guide***—For all users. This guide includes an introduction to TCPware for OpenVMS products as well as a reference for the user functions arranged alphabetically by product, utility, or service.
- ***Management Guide***—For system managers. This guide contains information on functions not normally available to the general network end user. It also includes implementation notes and troubleshooting information.
- ***Network Control Utility (NETCU) Command Reference***—For users and system managers. This reference covers all the commands available with the Network Control Utility (NETCU) and contains troubleshooting information.
- ***Programmer's Guide***—For network application programmers. This guide gives application programmers information on the callable interfaces between TCPware for OpenVMS and application programs.
- Online help—
  - Topical help, using **`HELP TCPWARE`** ***`[topic]`***
  - Error messages help, using **`HELP TCPWARE MESSAGES`**

Conventions Used

Convention	**Meaning**
host	Any computer system on the network. The local host is your computer. A remote host is any other computer.
`monospaced type`	System output or user input. User input is in `bold type`. Example: `Is this configuration correct? YES` Monospaced type also indicates user input where the case of the entry should be preserved.
italic type	Variable value in commands and examples. For example, *username* indicates that you must substitute your actual username. Italic text also identifies documentation references.
[*directory*]	Directory name in an OpenVMS file specification. Include the brackets in the specification.
[optional-text]	(Italicized text and square brackets) Enclosed information is optional. Do not include the brackets when entering the information. Example: `START/IP line` *`address [info]`* This command indicates that the *info* parameter is optional.
{*value* | *value*}	Denotes that you should use only one of the given values. Do not include the braces or vertical bars when entering the value.
Note!	Information that follows is particularly noteworthy.
CAUTION!	Information that follows is critical in preventing a system interruption or security breach.
`key`	Press the specified key on your keyboard.
`Ctrl/key`	Press the control key and the other specified key simultaneously.
`Return`	Press the Return or Enter key on your keyboard.

Chapter 1 Introducing TCPware for OpenVMS

Introduction

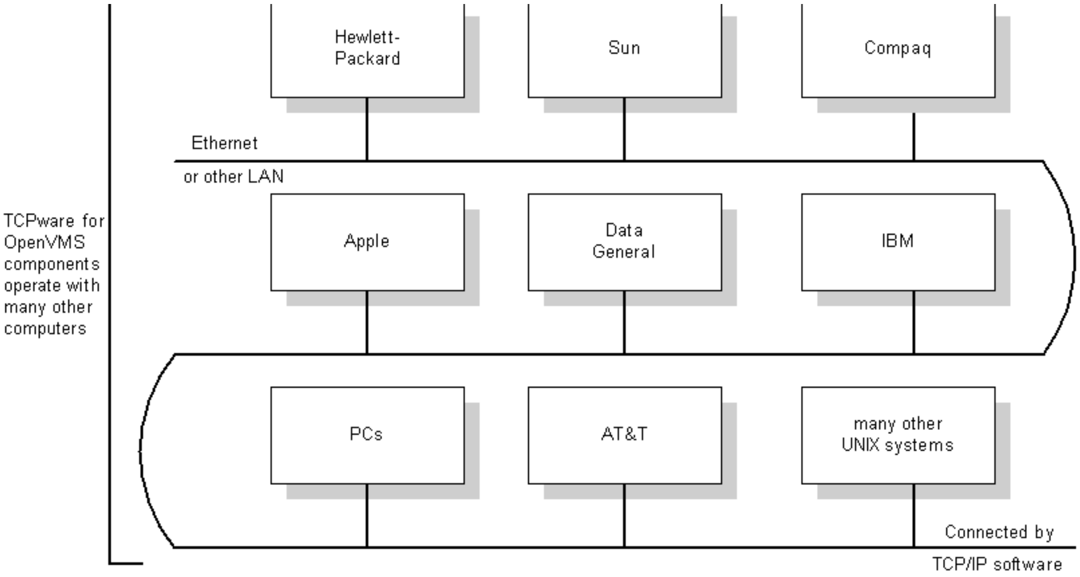
TCPware for OpenVMS is a software product that provides TCP/IP standard networking services for HP's OpenVMS VAX, Alpha and I64 computers.

Enterprise-Wide Networking

Computer systems from many different vendors can communicate with systems using the TCP/IP protocols. Almost all UNIX-based systems support TCP/IP, FTP, NFS, SMTP, and TELNET. This makes TCPware for OpenVMS components ideal tools for networking OpenVMS systems with other computer systems.

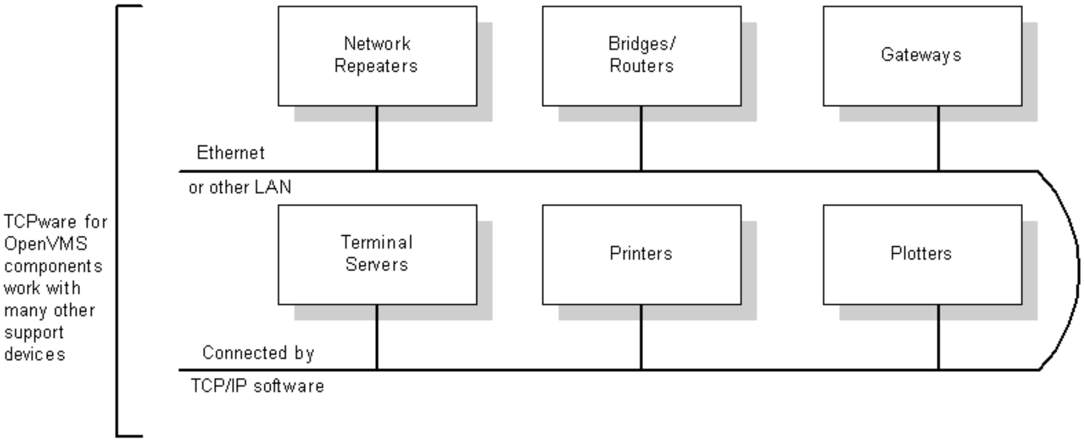
Figure 1-1 shows some systems networked using TCP/IP.

Figure 1-1 Connecting Dissimilar Systems Using TCPware for OpenVMS



TCPware for OpenVMS components operate with many other computers. TCPware for OpenVMS components also operate with many network support devices that are compatible with TCP/IP, Ethernet, and other local area networks (LANs), as shown in Figure 1-2.

Figure 1-2 Devices Supporting TCP/IP Networking



TCPware for OpenVMS

TCPware for OpenVMS includes the TCP/IP Services components designed exclusively for the VAX, Alpha and I64 architectures and the OpenVMS operating system for those architectures.

Table 1-1 lists the members of the TCPware for OpenVMS family and the features of each.

Table 1-1 TCPware for OpenVMS Family Members

Component	Features
FTP-OpenVMS	File transfer service that lets you transfer files to or from remote hosts. Provides a File Transfer Protocol (FTP) client and server. Includes the Remote Copy Program (RCP) (which includes optional Kerberos authentication). Also includes a Subroutine Library to develop FTP application programs. Token Authentication is also available for FTP-OpenVMS.
NFS-OpenVMS Client	Network File System (NFS) service that lets you access NFS filesystems and store data on NFS systems. Provides an NFS client.
NFS-OpenVMS Server	NFS service that lets remote NFS users access OpenVMS filesystems and use them for storage. Provides an NFS server and supports a PC-NFS Server (PCNFSD).
SMTP-OpenVMS	Mail transfer service that lets you send mail to or receive mail from remote hosts. Provides a Simple Mail Transfer Protocol (SMTP) client and server. The additional Internet Message Access Protocol (IMAP) and Post Office Protocol Version 3 (POP3) servers provide a way for remote PCs to retrieve OpenVMS incoming mail.
SSH-OpenVMS	Secure Shell provides encrypted remote access to this system and other systems with SSH software. Commands may be executed remotely or remote interactive sessions may be used. Files may be transferred with the SCP command, which uses SSH for access to the remote system.
TELNET-OpenVMS	Virtual terminal service that lets you have immediate access to remote systems. Provides a Virtual Terminal Networking (TELNET) protocol client and server. Kerberos authentication is also available. Also includes a Subroutine Library to develop TELNET application programs. Token Authentication is also available for TELNET-OpenVMS.

TCP-OpenVMS	TCP/IP base component that includes protocols for the network layer (IP, ICMP, ARP, and RARP) and transport layer (TCP and UDP). Provides utilities for network management and control: For ***Domain Name Services (DNS), Simple Network Management Protocol (SNMP) Services, Network Control Utility (NETCU),*** and ***Network Time Synchronization,*** see the *Network Management* entry in TCP/IP Services . • ***Berkeley R Commands*** — Access hosts in a TCP/IP network by logging in (RLOGIN), executing remote commands (RSH), and controlling remote tape drives (RMT) and CD-ROM drives (RCD). Kerberos V4 authentication is also available for RLOGIN and RSH. Token Authentication is also available for RLOGIN. • ***Line Printer Services*** — Manipulate local or remote print queue functions based on the client and server ends of the BSD4.3 Line Printer Protocol. • ***Terminal Server Print Services*** — Send print requests to printers attached to TCP/IP-based terminal servers. • ***Subroutine Libraries*** — Facilitate application development using the Socket Library Services, FTP Subroutine Library, TELNET Subroutine Library, and SNMP Extendible Agent Application Program Interface (API) routines. • ***TCPDRIVER, UDPDRIVER, IPDRIVER, and INETDRIVER Programming Services, and UCX Compatibility Services (BGDRIVER)*** — Use QIO interfaces to develop network applications. UCX Compatibility allows applications such as PATHWORKS to work with TCPware. ***ONC RPC Services***—Build distributed applications using Remote Procedure Calls (RPCs).

TCP/IP Services

All TCPware for OpenVMS TCP/IP Services are fully integrated. The services range from the upper-layer Network Application Services to the lower-level components. These lower-level components handle the network controllers included in the TCP/IP Services core component, TCP-OpenVMS.

The TCPware for OpenVMS components use the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Protocol (IP). The Department of Defense (DoD) adopted the IP and TCP protocols as standards for all packet networks. TCP and IP provide a reliable and efficient means for moving information between computer systems.

TCPware supports Path MTU discovery to provide a performance improvement when large packets of data are sent over TCP.

Table 1-2 describes some of the TCP/IP functions supported by TCPware.

Table 1-2 TCP/IP Services

This Service...	**Provides...**
Cluster Load Balancing	Have the domain name server assign a connection to a specific host to balance the cluster load. Analogous to the load balancing services the LAT terminal service provides.
Database Support	Connect Ingres, Oracle, RDB, Progress, and Sybase databases on OpenVMS and UNIX systems.
DECnet over IP	Send DECnet data link layer packets point-to-point over TCP/IP connection between two systems running TCPware.
DECwindows	Supports DECwindows graphics-oriented applications like Mail, File View, DECterm, and Bookreader. A remote X display user can also log in using the X Display Manager Server.

Interface Support	Interface support, which includes: • ***Ethernet, Token Ring, and LAT interfaces*** — Send IP datagrams over Ethernet, Token Ring, LAN Emulation over Asynchronous Transfer Mode (ATM), and Classical IP over ATM (CLIP) networks. Supports the Address Resolution Protocol (ARP) and Reverse ARP (RARP). • ***Fiber Distributed Data Interface (FDDI)* —** Send IP datagrams over high speed networks over FDDI controllers. Supports ARP and RARP. • ***HYPERchannel*** — Directly supports the UNIBUS, QBUS, MASSBUS, and BIBUS interfaces. Includes use of ARP to map host internet addresses to physical addresses. • ***IP-over-DECnet* —** Send IP datagrams over DECnet links to connect separate DECnet-over-IP TCP/IP LANs over WANs. • ***IP-over-X.25* —** Send IP datagrams as data packets over X.25, enabling reliable worldwide communication. • ***Point-to-Point Protocol (PPP)* —** Send multiprotocol datagrams over serial point-to-point links. PPP is common with line speeds from 14.4 to 28.8 kilobits per second (Kbps). Implemented through `pppd` command line options. • ***proNET* —** Supports the proNET-10 and proNET-80 token ring controllers provided by Proteon, Inc. • ***Serial Line IP (SLIP)* —** Send IP datagrams over serial lines instead of Ethernet cable. Supports both dedicated (hard-wired) and dialup SLIP lines. TCPware also supports ***Compressed SLIP (CSLIP)***. • ***HP Wide Area Network (WAN) Device Drivers* —** Supports the VAX WAN Device Drivers synchronous interfaces that form a link between the hardware devices and TCPware.
Multicasting	Supports full IP multicasting, letting you send and receive datagrams addressed to IP multicast (Class D) addresses. Implements the Internet Group Management Protocol (IGMP).

Network Management	Network management and control functions include: • ***Domain Name Services (DNS) —*** Guarantee host connections using a distributed database. Supports Berkeley Internet Domain Server (BIND) Release 4.9.4 Name Server. • ***Dynamic Host Configuration Protocol (DHCP) —*** Provides IP addresses and configuration data to hosts. Supports DHCP and BOOTP protocols. • ***Simple Network Management Protocol (SNMP) Services —*** Network management stations can obtain timely information about the network activities of OpenVMS server hosts. Supports MIB-I and MIB-II. TCPware's SNMP Agent also supports subagents serving private MIBs, as well as the SNMP Multiplexing (SMUX) Service. • **Network Control Process (NETCP) —** Starts, maintains, and shuts down the network. NETCP also contains the Port Mapper that maps Remote Procedure Call (RPC) server programs to ports. A TCPDUMP utility is also included. • ***Network Control Utility*** (NETCU) — Provides commands so that the system manager can monitor and control various functions such as adding and removing servers and clients. • ***Network Time Synchronization —*** Use either the Network Time Protocol (NTP) or the Time Synchronization Protocol (TIMED), to coordinate time distribution between hosts.
Network Security	Includes Incoming and Outgoing Access Restrictions; Packet Filtering; the Kerberos V4 Server, user commands, management commands, and administration server; the IP Security Option (IPSO); and Token Authentication for login security.
Other Clients and Servers	Client protocols (DISCARD, FINGER, NSLOOKUP, PING, TALK, Trivial File Transfer Protocol [TFTP], TRACEROUTE, and WHOIS) and Server protocols (CHARGEND, DAYTIMED, DISCARDED, ECHOD, FINGERD, INDENT, QUOTED, and TFTPD).
PATHWORKS Support	Use TCPware as a transport for HP's PATHWORKS products running between the OpenVMS system and a PC.
Routing	Supports enhanced routing and multiple gateways, and includes the GateD protocol, which combines RIP, HELLO, OSPF, EGP, BGP, and the Router Discovery Protocol for distributing routing information. Supports the Classless Inter-Domain Routing (CIDR) protocol for more efficient use of Class B IP addresses.

TCPware Products for the PDP-11 Operating Systems

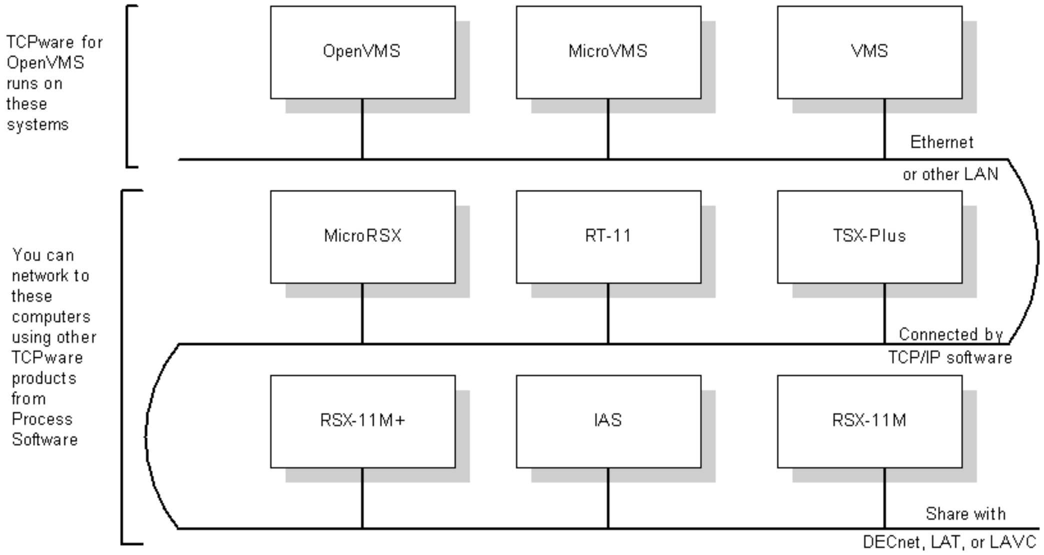
Process Software offers TCP/IP networking software products for the HP PDP-11 operating systems. Order the following TCPware products for the PDP-11 systems from Process Software:

TCPware for RSX	TCPware for RT-11	TCPware for TSX	TCPware for IAS

Although these products function differently from TCPware for OpenVMS, they solve many networking problems between dissimilar computer systems.

Figure 1-3 shows some of the many dissimilar systems TCPware can connect.

Figure 1-3 HP Operating Systems Connected by TCPware


TCPware for OpenVMS runs on these systems
OpenVMS
MicroVMS
VMS
Ethernet or other LAN
You can network to these computers using other TCPware products from Process Software
MicroRSX
RT-11
TSX-Plus
Connected by TCP/IP software
RSX-11M+
IAS
RSX-11M
Share with DECnet, LAT, or LAVC

Chapter 2 Functional Overview

Introduction

This chapter presents a functional overview of the TCPware for OpenVMS components. It addresses questions you may have, such as what you use to:

- Access to network filesystems as if they were local filesystems
- Transfer (copy) files over the network
- Print network files
- Log in to and perform commands on a remote system
- Send or receive mail or message over the network
- Access to network magnetic tape or CD-ROM drives
- Dynamically configure network hosts and find network information
- Control network activity
- Synchronize clocks across the network
- Secure resources on the network
- Tunnel external protocol applications over IP
- Program network interfaces

For more details on each subject, we provide you with references to the appropriate section of this documentation set at the end of this chapter.

Remote Filesystem Access

You can access remote filesystems as if they were your own, using NFS-OpenVMS (see Table 2-1).

Table 2-1 TCPware Components for Access to Network Filesystem

This component...	**Allows you to...**	**To use it, you need...**	**As a user...**	**As a system manager...**
NFS-OpenVMS Client	On a TCP/IP network, transparently access filesystems on remote servers so that they appear as resident filesystems in OpenVMS.	To access remote filesystems, run the NFS-OpenVMS Client. You must have authorization to access them.	simply use the filesystems as if they were on your local system. No special commands are required.	see the *Management Guide*, Chapter 13, *Managing NFS-OpenVMS Client*.
NFS-OpenVMS Server	Provide a service so that remote system users can access your local OpenVMS filesystems as if they were their own.	For remote systems users to access OpenVMS files on your system, run the NFS-OpenVMS Server. The remote user must have authorization to access your local filesystems.		see the *Management Guide*, Chapter 14, *Managing NFS-OpenVMS Server*.

Transferring Files

You can transfer files to or from your OpenVMS system using FTP-OpenVMS (which includes the RCP feature) or the TFTP feature of TCP-OpenVMS. Transfer files using the TCPware for OpenVMS components in Table 2-2.

Table 2-2 TCPware Components for Transferring Network Files

This component...	**Allows you to...**	**To use it, you need...**
FTP-OpenVMS	Copy, get, and put files to and from remote systems using the File Transfer Protocol (FTP). TCPware provides both the client function so that local users can transfer files to and from remote systems, and the server function so that remote users can transfer files from your local system. Login authentication security is available through Token Authentication.	The remote system must support FTP. **As a user**, see the *User's Guide*, Chapter 3, *FTP: Transferring Files*. **As a system manager,** see the *Management Guide*, Chapter 12, *Managing FTP-OpenVMS*. **As a system programmer**, see the *Programmer's Guide*, Chapter 7, *FTP Library*.
RCP	Use a UNIX-like command to copy files to and from remote systems right on the system command line. TCPware also provides the RCP server so that remote users can copy files to or from your system.	The server must support equivalents of the UNIX `shell` and `exec` services. You must register the other hosts in your `HOSTS.EQUIV` or `.RHOSTS` files. **As a user**, see the *User's Guide*, Chapter 7, *RCP: Copying Files*. **As a system manager**, see the *Management Guide*, Chapter 16, *Managing R Commands*.
TFTP	Transfer files to and from remote systems. Because TFTP is more primitive than FTP, you can mainly use TFTP to allow remote diskless systems to read bootstrap images over the network.	The remote system must support TFTP. **As a user**, see the *User's Guide*, Chapter 13, *TFTP: Trivial File Transfers*. **As a system manager**, see the *Management Guide*, Chapter 16, *Managing R Commands*.

Printing Files

You can print files over the network using the Line Printer Services or Terminal Server Print Services. Print files over the network using the TCPware for OpenVMS components in Table 2-3.

Table 2-3 TCPware Components for Network Printing

This component...	**Allows you to...**	**To use it, you need...**	**As a user...**	**As a system manager**...
Line Printer Services	Send files to, remove jobs from, and display the status of remote print queues using UNIX-like commands. Line Printer Services also provides a server so that remote users can access local print queues.	to define the remote printers during installation.	see the *User's Guide*, Chapter 5, *Networking Printing*.	see the *Management Guide*, Chapter 15, *Managing Print Services*.
Terminal Server Print Services	If you are on a TCP/IP network, send files to printers connected to remote terminal servers.	Use the regular PRINT/QUEUE commands.	see the *User's Guide*, Chapter 5, *Network Printing*.	see the *Management Guide*, Chapter 15, *Managing Print Services*, the *Terminal Server Print Services* section.

Logging In to Remote Hosts

You can log in to and execute commands on remote hosts using the RLOGIN or RSH features of TCP-OpenVMS or TELNET-OpenVMS. Log in to or emulate remote hosts using the components in Table 2-4.

Table 2-4 TCPware Components for Logging in to Remote Hosts

This component...	**Allows you to...**	**As a system programmer...**	**As a user...**	**As a system manager...**
RLOGIN	Use a UNIX-like command to log in to a remote host.		see the *User's Guide*, Chapter 8, *RLOGIN: Logging In to a Remote Host*.	see the *Management Guide*, Chapter 16, *Managing R Commands*.
RSH	Use a UNIX-like command to execute a single command on a remote host without logging in.		see the *User's Guide*, Chapter 9, *RSH: Issuing Commands on the Remote Host*.	see the *Management Guide*, Chapter 16, *Managing R Commands*.
TELNET-OpenVMS	Initiate virtual terminal connections to remote hosts using the TELNET protocol. You can open multiple remote sessions. TCPware also provides a server function so that remote users can make virtual terminal connections to the OpenVMS host. Login authentication security is available through Token Authentication.	see the *Programmer's Guide*, Chapter 9, *TELNET Library*.	see the *User's Guide*, Chapter 12, *TELNET: Connecting to Remote Terminals*.	see the *Management Guide*, Chapter 18, *Managing TELNET-OpenVMS Server*.

Transferring Mail and Exchanging Messages

You can send and receive mail over the network using the TCPware for OpenVMS components in Table 2-5.

Table 2-5 TCPware Components for Sending Network Mail

This component...	**Allows you to...**	**To use it, ...**	**As a user...**	**As a system manager...**
SMTP-OpenVMS	On a TCP/IP network, send and receive mail over the network using the Simple Mail Transfer Protocol (SMTP). TCPware provides both an SMTP client and a server.	The remote system must support SMTP.	see the *User's Guide*, Chapter 10, *SMTP: Transferring Mail*.	see the *Management Guide*, Chapter 17, *Managing Mail Services*.
IMAP Server	Provide a service so that remote PCs can access mail in VMS MAIL mailboxes using the Internet Message Access Protocol (IMAP) Server.	The remote system must support the IMAP protocol.		see the *Management Guide*, Chapter 17, *Managing Mail Services*, the *IMAP Server* section.
POP3 Server	Provide a service so that remote PCs can retrieve mail in VMS MAIL in-boxes using the Post Office Protocol (POP3) Server.	The remote system must support the POP3 protocol.		see the *Management Guide*, Chapter 17, *Managing Mail Services*, the *POP3 Server* section.

TALK Utility	Exchange "real time" messages with another host on the local or remote network. Display simultaneously sent and received messages on a split screen.	The remote system must support the `talk` protocol.	see the *User's Guide*, Chapter 11, *TALK: Exchanging Terminal Messages*.	

Accessing Network Drives

You can access remote tape or CD-ROM drives, or provide access locally to remote users by using the TCPware for OpenVMS components in Table 2-6.

Table 2-6 TCPware Features for Providing Access to Network Tape Drives

This component...	**Allows you to...**	**To use it, you need to configure...**	**As a user...**	**As a system manager...**
RMT Client	Use OpenVMS commands such as BACKUP, MOUNT, COPY, and EXCHANGE on remote backup tape drives.	a pseudo-device on your OpenVMS system using the command RMTSETUP. The remote system must support the `rmt` protocol.	see the *User's Guide*, Chapter 6, *RCD and RMT: Remote CD-ROMs and Tapes*.	
RCD Client	Use OpenVMS commands such as BACKUP, MOUNT, COPY, and EXCHANGE on remote CD-ROM drives.	a pseudo-device on your OpenVMS system using the command RMTSETUP. The remote system must support the `rmt` protocol.	see the *User's Guide*, Chapter 6, *RCD and RMT: Remote CD-ROMs and Tapes*.	
RMT Service	Provide a	the Berkeley R		see the

	service so that remote clients can use the `rdump` or `rrestore` UNIX utilities to access a magnetic tape on your system.	Commands for RMT services. The remote system must support the `rmt` protocol.		*Management Guide,* Chapter 16, *Managing R Commands.*

Configuring Hosts

TCPware provides various components and features with which you can configure network hosts, as listed in Table 2-7.

Table 2-7 TCPware Features for Configuring Hosts

This component...	**Allows you to...**	**As system manager...**
DHCP/BOOTP	Assign IP addresses and provide configuration data to hosts over the network.	see the *Management Guide*, Chapter 2, *DHCP/BOOTP Server*.
Domain Name Services	Obtain information such as host Internet addresses and names by connecting to a distributed database.	see the *Management Guide*, Chapter 3, *Domain Name Services*.
Point-to-Point Protocol (PPP)	Configure the network to send IP datagrams over serial links, including DECnet or modern connections.	enter: `$ PPPD:==TCPWARE:PPPD.EXE` See the *Management Guide*, Chapter 5, *Serial Link Interfaces: PPP and SLIP*.
Serial Line IP (SLIP) Protocol	Further configure the network to send IP datagrams over serial links.	

Controlling Network Functions

You can perform network management functions and test networks by using the TCPware for OpenVMS features in Table 2-8.

Table 2-8 TCPware Features for Additional Management

This component...	**Allows you to...**	**As a system manager...**
Network Control Utility (NETCU)	NETCU is the utility program system managers and user use to configure and control network activity.	see the *NETCU Command Reference*.
Simple Network Management Protocol (SNMP) Services	Obtain timely information about network activities of OpenVMS server hosts, such as routing, line status, volume of traffic, and error conditions. SNMP supports the MIB-I and MIB-II Management Information bases, as well as SNMP Multiplexing (SMUX) and SNMP Agent eXtensibility (AGENTX).	see the *Management Guide*, Chapter 7, *Managing SNMP Services*. see the *Programmer's Guide*, Chapter 10, *SNMP Extendible Agent API Routines*.

Synchronizing Time Clocks

TCPware provides the network time synchronization components listed in Table 2-9.

Table 2-9 TCPware Features for Time Synchronization

This component...	**On a TCP/IP network, allows you to...**	**As a system manager, see the *Management Guide*,**
Network Time Protocol	synchronize your system clock with an Internet Time Server.	Chapter 10, *Network Time Protocol (NTP).*
TIMED	use the Time Synchronization Protocol (TSP) and the `timed` service to synchronize the clocks of LAN hosts.	Chapter 11, *TIMED.*

Using Network Testing Tools

TCPware provides various network testing tools, and utilities and services with which you can obtain network information, as listed in Table 2-10.

Table 2-10 TCPware Network Testing Tools

This component...	Allows you to...	As a user...	As a system manager...
FINGER	Extract user information from a remote user information program.		enter: `$ FINGER user@host-to-finger` See the *Management Guide*, Chapter 30, *Network Testing Tools*.
IDENT	Determine the user associated with a connection.		See the *Management Guide*, Chapter 30, *Network Testing Tools*.
NSLOOKUP	Extract information about network hosts from the Domain Name Systems.		enter: `$ nslookup host-to-find` See the *Management Guide*, Chapter 30, *Network Testing Tools*.
PING	Find out if a host is up and if you can reach it.		enter: `$ PING:==$TCPWARE:PING` See the *Management Guide*, Chapter 30, *Network Testing Tools*.
TCPDUMP Utility	Track TCP packets by printing information in packet headers.		See the *Management Guide*, Chapter 30, *Network Testing Tools*.
TRACEROUTE	Trace the path of an IP packet to an internet host.		See the *Management Guide*, Chapter 30, *Network Testing Tools*.

WHOIS	Query the Network Information Center (NIC) username directory services to obtain usernames.	enter the command: `$ WHOIS username` See the *User's Guide*, Chapter 15, *WHOIS: Username Directory Services*.	

TCPware also provides other useful testing utilities and services, including CHARGEND, DAYTIMED, DISCARD, ECHOD, NETCU DEBUG, QUOTED, and TIME. See the *Management Guide*, Chapter 31, *Network Testing Tools*, for details.

Securing Resources

You can secure resources on the network using the TCPware features described in Table 2-11.

Table 2-11 TCPware Features for Securing Network Resources

This component...	**Allows you to...**	**As a system manager, see the *Management Guide*,**	**As a user, see the *User's Guide*,**
Incoming Access Restrictions	Restrict the hosts and networks that can access the services the master server activates.	Chapter 20, *Access Restrictions*.	
Outgoing Access Restrictions	Restrict requests for remote services to specific users and ports.	Chapter 20, *Access Restrictions*.	
Packet Filtering	Restrict the datagrams a network interface can receive by protocol, source and destination address, or destination port. Use convenient NETCU commands.	Chapter 21, *Packet Filtering*.	

Kerberos Server	Provide password encryption and the Key Distribution Center (KDC) for getting tickets to server applications. Also use management and user commands.	Chapter 23, *Managing Kerberos*.	Chapter 4, *Kerberos User Commands*.
Kerberos Authentication for RCP	Use Kerberos V4 authentication with the RLOGIN Berkeley R Command.	Chapter 23, *Managing Kerberos*.	Chapter 7, *RCP: Copying Files*.
Kerberos Authentication for RLOGIN	Use Kerberos V4 authentication with the RLOGIN Berkeley R Command.	Chapter 23, *Managing Kerberos*.	Chapter 8, *RLOGIN: Logging in to a Remote Host*.
Kerberos Authentication for RSH	Use Kerberos V4 authentication with the RSH Berkeley R Command.	Chapter 23, *Managing Kerberos*.	Chapter 9, *RSH: Issuing Commands on the Remote Host*.
Kerberos Authentication for TELNET	Use Kerberos V4 authentication with TELNET-OpenVMS.		Chapter 12, *TELNET: Connecting to Remote Terminals*.
IP Security Option (IPSO)	Provide IP datagram protection using the IP Security Option (IPSO) protocol.	Chapter 24, *IP Security Option (IPSO)*.	

Token Authentication	Use a Security Dynamics "smart card" token and TCPware's ACE/Client and its use of the ACE/Server to authenticate logins from FTP-OpenVMS, TELNET-OpenVMS, RLOGIN, and SET HOST sessions.	Chapter 22, *Managing Token Authentication*.	Chapter 14, *Token Authentication: Protecting Logins*.
Secure Shell (SSH)	Configure and maintain the TCPware Secure Shell (SSH) server. This is the server side of the software that allows secure interactive connections to other computers in the manner of rlogin/rshell/telnet.	Chapter 25, *Configuring the Secure Shell (SSH) Server*.	Chapter 16, *Accessing Remote Systems with the Secure Shell (SSH) Utilities*

Tunneling External Applications over IP

You can tunnel DECnet applications over IP networks if you are using DECnet Phase IV (see Table 2-12.) A connection established between two systems running different protocols is known as a tunnel.

Table 2-12 TCPware Features for Tunneling Applications over IP

This component...	**Allows you to...**	**As a system manager, see the *Management Guide*,**
Tunneling DECnet over IP (for DECnet Phase IV)	Connect two DECnet networks over an IP link. Use with DECnet Phase IV only. There is no need to use this feature with DECnet/OSI (DECnet Phase V).	Chapter 28, *Tunneling DECnet over IP*.

Programming Network Interfaces

If you are a network programmer, you can perform programming functions using the programming interfaces discussed in the Programmer's Guide (see Table 2-13).

Table 2-13 TCPware Network Programming Interfaces

This component...	**Allows you to...**	**As a system programmer, see the *Programmer's Guide*,**
FTP Library	Use a programming interface to the FTP protocol. Use the FTP-OpenVMS library routines in your own applications to provide FTP capabilities.	Chapter 7, *FTP Library*.
Socket Library	Use either the HP Computer C Socket Library (for OpenVMS Version 5.3 and later) or the TCPware Socket Library (for earlier version or you are using the Remote Procedure Call routines).	Chapter 8, *Socket Library*.
TELNET Library	Use a programming interface to the TELNET protocol. Use the TELNET-OpenVMS library routines in your own applications to provide FTP capabilities.	Chapter 9, *TELNET Library*.
UCX Compatibility Services	Use the BGDRIVER $QIO programming interface for compatibility with HP's TCP/IP Services for OpenVMS (formerly UCX) product.	Chapter 2, *UCX Compatibility Services*.
QIO Programming Interfaces	Use $QIO programming interfaces to TCP/IP. These include the BGDRIVER, TCPDRIVER, UDPDRIVER, IPDRIVER, and INETDRIVER interfaces.	Chapter 10, *SNMP Extendible Agent API Routines*.
SNMP Extendible Agent Application Programming Interface (API) Routines	Use API routines required for an application program to export private Management Information Bases (MIBx) using the TCPware SNMP agent.	Chapter 10, *SNMP Extendible Agent AAPI Routines*.

Token Authentication ACE/Client API Functions	Use API functions for programs that interact between the ACE/Client and ACE/Server to enable Token Authentication.	Chapter 11, *Token Authentication API Functions*.

Chapter 3 FTP: Transferring Files

Introduction

The File Transfer Protocol (FTP) transfers files to and from a remote host. FTP-OpenVMS controls the method by which FTP transfers the files.

The Client-FTP utility is your interface to FTP-OpenVMS. You can run Client-FTP interactively or through a startup command procedure.

For FTP-OpenVMS to operate between two hosts, the remote host must provide a compliant client or server. You can run FTP directly (interactively) or indirectly from a command procedure. Client-FTP supports multiline recall of up to 20 lines.

Before Using FTP

Before you can transfer files, you need:

- To make sure that the FTP-OpenVMS software is installed, configured, and started on your system.
- The name or internet address of the remote host to which you want to connect.
- The username and password of the account on the remote host. If the remote host does not support multiuser protection features, you might not need a username and password. If you are using TCPware's Token Authentication, the password is the PASSCODE generated on your SecurIDW token.
- The filenaming conventions on the remote host.

FTP Session

A typical FTP session consists of the following steps:

1. Open the FTP connection.
2. Determine the format of the files you want transferred.
3. Transfer files using the GET (MGET), PUT (MPUT), or COPY commands or selections on the graphical user interface windows. The default file format is formatted ASCII.
4. Close or exit the FTP connection.

Features

FTP-OpenVMS includes the following features:

- Choice of command line execution or graphical user interface execution (for DECwindows Motif Version 1.1 or later).

- Informational and error status messages.
- Support of wildcards in source filespecs.

Table 3-1 describes some of the features of Client-FTP.

Table 3-1 Client-FTP Features

This feature...	**Means that...**
Command Line or Graphical User Interface Command Execution	Client-FTP allows you to execute FTP commands either at the `FTP>` prompt or through a DECwindows graphical user interface environment. The user interface is provided with DECwindows Motif Version 1.1 and later. You can use either DCL-style syntax or UNIX-style syntax at the `FTP>` prompt. DCL-syntax can include qualifiers: `FTP>`**`DIRECTORY *.DIR /BRIEF`** You usually enter UNIX-style commands in lowercase: `FTP>`**`ls *.dir`**
Case Conversion	Client-FTP no longer converts the user name, password, and account to lowercase if they were not supplied on the OPEN and USER command line and thus prompted for. If you are prompted for these parameters you must enter them in the proper case, since quotes are no longer needed to maintain case.
Status Messages	Client-FTP issues informational and error messages. These messages are self-explanatory and conform to the standard OpenVMS message format. The numeric codes that prefix these messages conform to the RFC 959 standard for FTP.

Wildcards	Client-FTP supports wildcards for the COPY, GET, PUT, DELETE, and DIRECTORY commands. The acceptable wildcard characters are: • Percent sign (%) or question mark (**?**) to represent individual characters. • Asterisk (*) to represent multiple characters. If you include the asterisk wildcard to represent multiple files to FTP, use the MGET, MPUT, or MDELETE commands, or specify the /MULTIPLE qualifier with the GET, PUT, COPY, or DELETE command. These two examples produce identical results: `FTP>`**`MGET *.TXT`** `FTP>`**`COPY *.TXT/MULTIPLE/REMOTE *`** ***Note!*** You do not require the asterisk for the destination with MGET, but you do require it with COPY. If enclosed in a quoted string, wildcard symbols no longer act as wildcards.

Note! You can customize the appearance of your graphical user interface by using Motif resources in a resource file. This file is called DECW_FTP.DAT and is in your login directory.

The most important resource is the one that sets your application window to fit the screen. If you run your application from a PC with a small, 14-inch monitor, for example, you might want to use the following resource:

```
*DXmfitToScreenPolicy: AS_NEEDED
```

If the window size is bigger than the screen can handle, scroll bars appear in the windows so that you can scroll to parts of the window.

Other examples of using resources include:

```
DECW_FTP*background: gray

DECW_FTP*foreground: black
```

These set the screen background color to gray and the foreground color to black. See your Motif documentation for other possible resource settings.

Note! Wherever possible, the procedural descriptions that follow cover the command line and graphical user interface execution methods. If you prefer the graphical user interface method, you can execute most file transfer and manipulation functions from the **`TCPware FTP-OpenVMS File Transfers`** window shown in File Transfers Window and FTP-OpenVMS Window Options. Many of the functions in this window have command line equivalents.
If you need further information on performing a particular function in the **`TCPware FTP-OpenVMS File Transfers`** window, see its command equivalent in the *Command Reference*.

Note! TCPware provides secure FTP-OpenVMS logins through its Token Authentication feature, if installed and enabled. For more information, see Chapter 14, *Token Authentication: Protecting Logins.*

Opening a Connection

Only one FTP connection can be open at a time. Once open, all file transfers and other remote operations use that connection.

You can open an FTP connection by using either the command line user interface, or the graphical user interface if you have a DECwindows system.

Command line method. Use this method if you want to issue commands from the DCL prompt (see Figure 3-1 Opening an FTP Connection Using the Command Line Method).

1 Enter one of the following at the DCL prompt:

```
$ FTP
FTP>OPEN host
```

in combination:

- *host* is the name of the host to which you want to connect. Respond to the login prompts, if any, of the remote host. After a successful login, the `FTP>` prompt appears where you enter the FTP commands described in the following sections. This is the option shown in Figure 3-1.

  $ **FTP[/TLS]** ***host***

- *host* is the name of the host to which you want to connect. Respond to the login prompts, if any, of the remote host. After a successful login, the `FTP>` prompt appears where you enter the FTP commands described in the following sections. If /TLS is included on the command line, then TLS authentication will be used before user authentication is entered.

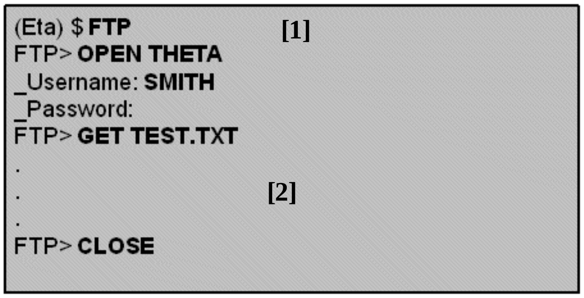
  $ **FTP[/TLS]** ***host username password***

  Enter the host to which you want to connect, the username of the account on the remote host, and the password (PASSCODE if using Token Authentication) of the account on the remote host as part of the command. After a successful login, the `FTP>` prompt appears where you enter the FTP commands described in the following sections.

See the OPEN command if you are using a SecureID card for password authentication.

2 Closing and Exiting for the different close options.)

Figure 3-1   Opening an FTP Connection Using the Command Line Method



```
(Eta) $ FTP
FTP> OPEN THETA
_Username: SMITH
_Password:
FTP> GET TEST.TXT
.
.
.
FTP> CLOSE
```

Graphical user interface method. You can use the graphical user interface method if you have a DECwindows host running DECwindows with Motif Version 1.1 or later (see Figure 3-2):

1 At the DCL prompt, enter:

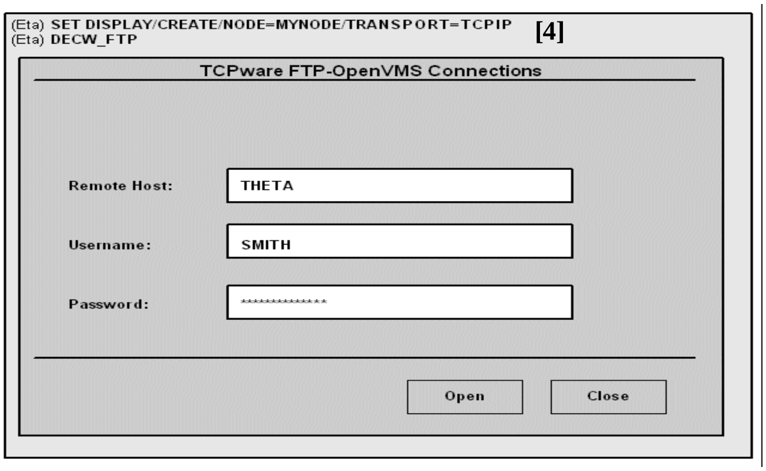
```
$ SET DISPLAY/CREATE/NODE=display-node/TRANSPORT=TCPIP
$ DECW_FTP
```

2 When the **`TCPware FTP-OpenVMS Connections`** window appears, enter at the **`Remote Host`**: field, tab to the **`Username`**: field and enter at it, and tab to the **`Password`**: field and enter at it. Then click the **`OPEN`** button.

A **`TCPware FTP-OpenVMS Message Window`** shows all the actions FTP-OpenVMS takes from this point on. Figure 3-3 shows an example of the **`TCPware FTP-OpenVMS File Transfers`** window that appears when you open a connection from the **`TCPware FTP-OpenVMS Connections`** window.

Note! For the graphical user interface, FTP-OpenVMS stores the connection information in the DECW_FTP_PROFILE.DAT file in your login directory to set up the next connection. See the note in the previous section first.

Figure 3-2 Opening an FTP Connection Using the Graphical User



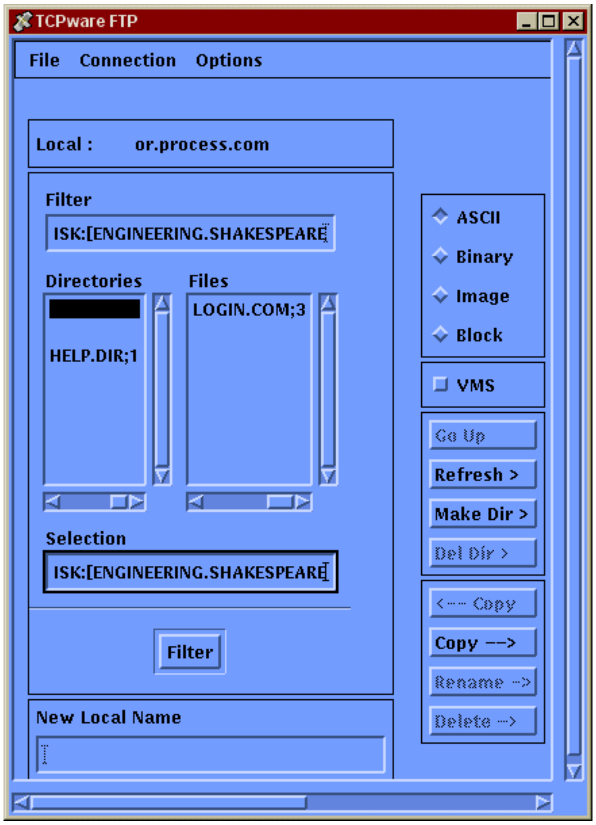
```
(Eta) SET DISPLAY/CREATE/NODE=MYNODE/TRANSPORT=TCPIP
(Eta) DECW_FTP
```

Remote Host: THETA

Username: SMITH

Password: **************

Figure 3-3 File Transfers Window



Local : or.process.com

Filter

ISK:[ENGINEERING.SHAKESPEARE

Directories

HELP.DIR;1

Files

LOGIN.COM;3

Selection

ISK:[ENGINEERING.SHAKESPEARE

New Local Name

☐ VMS

Graphical User Interface

The graphical user interface method offers a number of options from the **TCPware FTP-OpenVMS File Transfers** screen.

You can set various options by clicking **Options** on the menu bar on the **TCPware OpenVMS File Transfers** screen (see Figure 3-3). These options are:

- Settings (see the top screen in Figure 3-4)
- Viewer Preferences (see the bottom screen in Figure 3-4)

Here is the process to use:

1 Click **Options** followed by **Settings...** to get the **TCPware FTP-OpenVMS Settings** window. This window presents the following options:

FTP Logs	You can select to log **Commands, Replies**, or **Both**. Your password appears on the screen if you use the **Commands** or **Both** setting**. [1a]**
Confirm on Delete	Click the box to confirm file or directory deletion. **[1b]**
Beep After Copy	Click the box to enable a beep when copying is complete. **[1c]**
Timeout (secs)	Set the FTP session timeout, in seconds. **[1d]**
PASV Mode	Click the box to set passive mode transfers (see theSET command). **[1e]**
OK/Cancel	To accept the settings you make on this screen, click **OK**; to cancel the window, click **Cancel**. **[1f]**

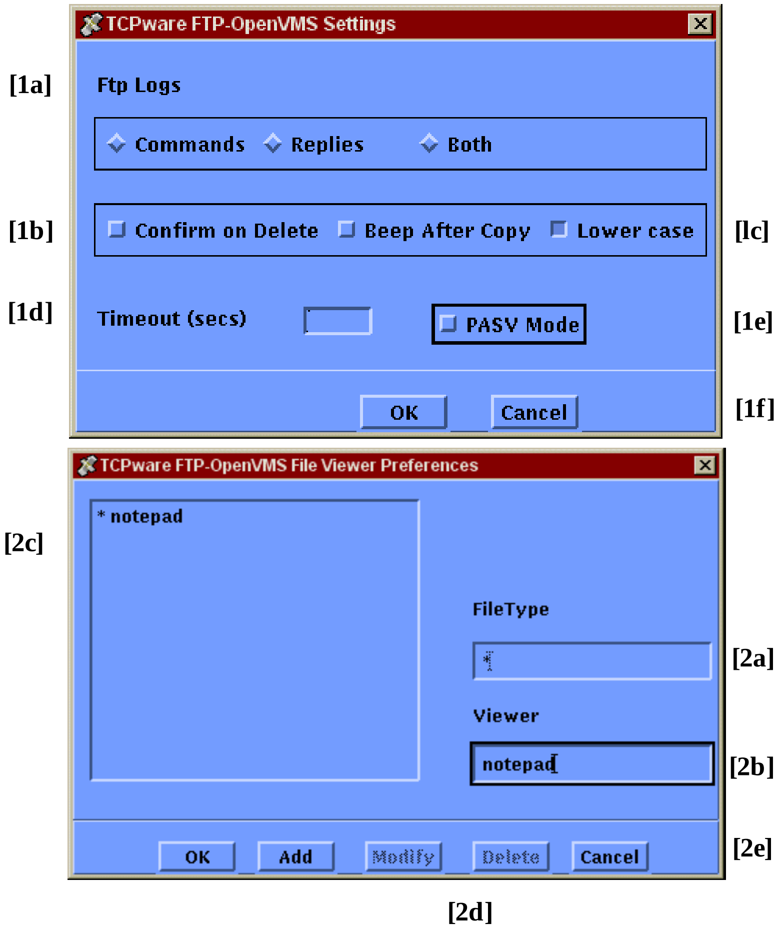
2 Click **Options** followed by **View...** to get the **TCPware FTP-OpenVMS File Viewer Preferences** window, with the following options:

File Type	Enter a file extension to indicate the type of file you would like to view; for example, enter **c** for files with the .C extension, **ps** for files with the .PS extension, or ***** for any file type. **[2a]**
Viewer	Enter the type of viewer to use for the file type; these should be DCL commands or foreign commands you define before invoking the application; for example, enter the DCL command **view/interface=decwindows/format=ps** to use the CDA Viewer with .PS files. **[2b]**
Add	Click **Add** to add the File Type and Viewer combination entered; the results appear in the scrollable list to the left of the **File Type** and **Viewer** fields. **[2c]**

Modify/ Delete	Click a list item and click the `Modify` or `Delete` button to modify or delete the item. **[2d]**
Cancel	To cancel the window, click `Cancel`. **[2e]**

Note! Changes you make to settings and viewer preferences are stored in DECW_FTP_SETTINGS.DAT and DECW_FTP_VIEWERS.DAT files, respectively, in your login directory.

Figure 3-4 FTP-OpenVMS Window Options



Closing and Exiting

An FTP connection remains open until you quit or exit FTP, close the connection, or open a new connection.

Command line method. See Figure 3-5:

1 To close an FTP connection, use one of the following commands:

FTP>**CLOSE**

Closes the current connection and continues the FTP session for the next command.

```
FTP>OPEN host
FTP> CONNECT host
```

Both OPEN and CONNECT close the current connection and open another one.

2 To exit an FTP session:

```
FTP>EXIT (or Ctrl/Z)
```

See the CLOSE, OPEN, and EXIT commands in the *Command Reference*.

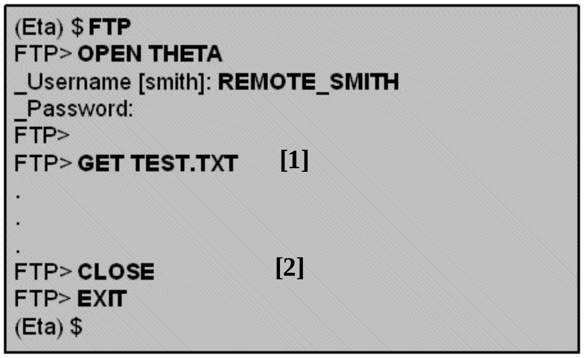
Graphical user interface method. See GET, PUT, and COPY Command Format:

1 To close an FTP connection from the `TCPware FTP-OpenVMS File Transfers` window, click the `Connections` option on the menu bar and click the `Close...` option. The information in the "Remote" part of the screen disappears.

To reopen a connection, click the `Connection` option on the menu bar and click the `Open...` option. (`Open` is initially greyed-out.)

2 To exit from FTP entirely from any of the DECwindows screens, click the `File` option on the menu bar and click the `Exit` option.

Figure 3-5 Closing from the Command Line



```
(Eta) $ FTP
FTP> OPEN THETA
_Username [smith]: REMOTE_SMITH
_Password:
FTP>
FTP> GET TEST.TXT      [1]
.
.
.
FTP> CLOSE             [2]
FTP> EXIT
(Eta) $
```

Checking Directories

After you establish an FTP connection, you can check the directories on the remote or local host to locate the file(s) you want.

To check remote directories and determine the file format type when in FTP (see Figure 3-6):

1 Open the FTP connection and enter: `FTP>DIRECTORY`

- Use the CD or SET DEFAULT /REMOTE command to move to other directories on the remote host.
- If you use the menu-driven method, see the "Remote" part of the `FTP for TCPware for OpenVMS` screen (see Figure 3-3). You can double-click any of the listed directories, change the pathname in the `Current Remote Directory` field, or use the `Go Up` button in the middle of the screen.

2 Check file extensions to determine file types. You might need to enter special qualifiers when you transfer certain types of files.

See Table 3-2 in the next section for a description of the file transfer formats.

3 Check the local directory when in FTP: `FTP>`**`LDIR`**

If you use the menu-driven method, see the "Local" part of the **`FTP for TCPware for OpenVMS`** screen (see Figure 3-2).

4 Use the LCD or SET DEFAULT /LOCAL command to move to other directories on the local host.

See the DIRECTORY, LDIR, and SET DEFAULT commands in the *Command Reference* for checking directories.

Figure 3-6 Checking Remote and Local Directories

```
FTP> DIRECTORY     [1]
total 49
-rwxr-xr-x    1  smith    users     340   Oct    1   16:34   .login
-rwxr-xr-x    1  smith    users     138   Oct    1   16:34   .profile  [2]
drwxr-xr-x     2  smith   users     512   Oct    1   16:34   bin
-rw-r--r--     1  smith   users   46080   Oct    1   10:58   sys.exe
drwxr-xr-x     2  root   daemon     512   Feb  10    2001   wastebaske
FTP> LDIR    [3]
Directory DOC$DISK:[DOC.ENG]
ANDY.TXT;1         CYN.PS;2     DO_HELP.TXT;1
GLOSSARY.TXT;1    HELP.DIR;1   KIT_INFO.PS;1
LWK_PERSONAL_LINKBASE;1        SCREEN-FTP.DIR;1 SEND-NORM.C;1
Total of 9 files.

FTP> LCD [.HELP]    [4]
FTP> LDIR
Directory DOC$DISK:[DOC.ENG.HELP]
BUILD.COM;1          FTPHELP.HLB;2         FTPHELP.RNO
FTPHELP.RNO;1        HELP.MMS;1

Total of 6 files.
```

Checking File Transfer Formats

You can determine what file format to use during file transfers. Client-FTP lets you transfer files in formatted ASCII, formatted binary, image, block, FORTRAN carriage control, and VMS formats. On OpenVMS systems, the filename extension can indicate the file type. Formatted ASCII is the default transfer file type and is usually sufficient for most files.

FTP converts the various file formats to formatted ASCII or IMAGE. (Executable and zip/compressed files are popular files in this category.) The formats are similar to the formats that the OpenVMS EXCHANGE utility provides to transfer between OpenVMS and DOS-11 or RT-11 file systems. You either specify the file transfer format when you use the GET, PUT, or COPY command, or Client-FTP determines the format from the source filename's extension.

See Figure 3-8 for an explanation of the file transfer formats.

Check file extensions to determine file types. You might need to enter special qualifiers when you transfer certain types of files.

- When you use the COPY, GET, or PUT commands to transfer files, you can use the /ASCII, /BINARY, /BLOCK, /FORTRAN, /IMAGE, or /VMS qualifiers to set the file transfer format. You can also set default file transfer formats using these qualifiers with the SET DEFAULT command, or specifying these keywords with the TYPE command. (See the SET DEFAULT and TYPE commands in the *Command Reference* for equivalent usage.)
- If you use the menu-driven method, you can make the file type selections in the middle part of the **`TCPware FTP-OpenVMS File Transfers`** screen (see Figure 3-4).

Table 3-2 Client-FTP File Transfer Formats

This file format...	With extension...	Means...
Formatted ASCII		ASCII records terminated with a CR and LF and transferred as ASCII. Use for all except formatted binary and image files. Maximum formatted ASCII record size is 8192 bytes. In OpenVMS-to-"FTP ASCII" conversion, CR/LF pairs are added to the end of records. In "FTP ASCII"-to-OpenVMS conversion, CR/LF pairs are removed from the end of records.
Formatted Binary	.OBJ .STB .BIN .LDA	Binary records transferred as IMAGE. In OpenVMS-to-"FTP IMAGE" conversion, record header and checksum are added to all records. In "FTP IMAGE"-to-OpenVMS conversion, record header and checksum are removed from each record. Remote hosts might not be able to distinguish between formatted binary and image files because both file types are transferred using "FTP IMAGE" format. In this case, the formatted binary files are stored as image files (and if properly transferred back, are formatted binary files again). This is typically not a problem because formatted binary files are system-dependent files.
BLOCK		File blocks transferred as IMAGE. Use for STREAM, STREAM_CR, STREAM_LF, and UNDEFINED record formats. Provides the highest transfer rates since it involves minimal processing. Very similar to image mode. In OpenVMS-to-"FTP IMAGE" conversion, and OpenVMS file is read using block-I/O mode without regard to record structure. In "FTP IMAGE"-to-OpenVMS conversion, an OpenVMS file is created with the STREAM_LF record format and is written using block-I/O mode. ***Note!*** No padding of the last block of data occurs. Block mode is particularly useful for files with a STREAM, STREAM_CR, STREAM_LF, or UNDEFINED record format.
FORTRAN		Like formatted ASCII except that first character of each line controls how to display each line. Conversions are the same as for formatted ASCII. Attributes for the output file reflect that the file has a FORTRAN carriage control format. Some hosts do not distinguish between FORTRAN carriage control and ASCII files and might not support this transfer format.

IMAGE	.EXE .TSK .OLB .MLB .SYS .SML .ULB	Fixed-length binary records transferred as IMAGE. In OpenVMS-to-"FTP IMAGE" conversion, records are read as is. In "FTP IMAGE"-to-OpenVMS conversion, records are written as fixed length. If the last record is too short (less than 512 bytes), it is padded with binary zeros.
VMS		Use for RMS file transfers between OpenVMS systems. Systems that support this structure negotiate it automatically. The VMS file structure types are richer than those of UNIX for which FTP is designed. Thus VMS and VMS-Plus modes were added to help in transferring OpenVMS files.

Using GET, PUT, and COPY

Use the GET, PUT, or COPY commands to transfer files.

GET	"Gets" a copy of a file from the remote host and places it in the current local directory.
PUT	"Puts" a copy of a local file in the current directory on the remote host.
COPY	"Gets" or "puts" a copy of a file, depending on use of the /LOCAL or /REMOTE qualifier after the source or destination parameter. COPY requires the destination parameter.

Command line method. Figure 3-7 shows the format and filename syntax of the GET, PUT, and COPY commands. Follow the examples and observe the following conventions when you transfer files between remote and local hosts (the sequence is not important):

- If using GET or PUT, omit *destination* if you want to use the *source* filename (and extension if it exists), unless *source* is a quoted string. COPY requires the destination parameter. If using COPY, use a wildcard (asterisk) for *destination* when you want to use the source filename as the destination filename.
- If copying to or from a non-OpenVMS filespec, enclose it in double quotes (" ").
- Separate multiple filespecs with commas.
- If using wildcarded source filespecs (with an asterisk), use the /MULTIPLE qualifier. Alternatively, use the MGET or MPUT command to copy wildcarded source files. (Note that this requires setting the remote default directory first.)
- Including an asterisk (*) after the semicolon (;) in a destination parameter preserves the file version when copying to a remote host.

Note! If the file version in the source parameter already exists at the destination, that version is overwritten at the destination. Also, you do not get a warning if a higher numbered destination version already exists.

- If a DECnet file, use the full OpenVMS filespec.
- At this point, the file transfer format you determined is important.

See the GET, PUT, and COPY commands in the *Command Reference*. The RCP command is also available at the DCL prompt for remote file copies (see Chapter 7, *RCP: Copying Files*, for details on its use).

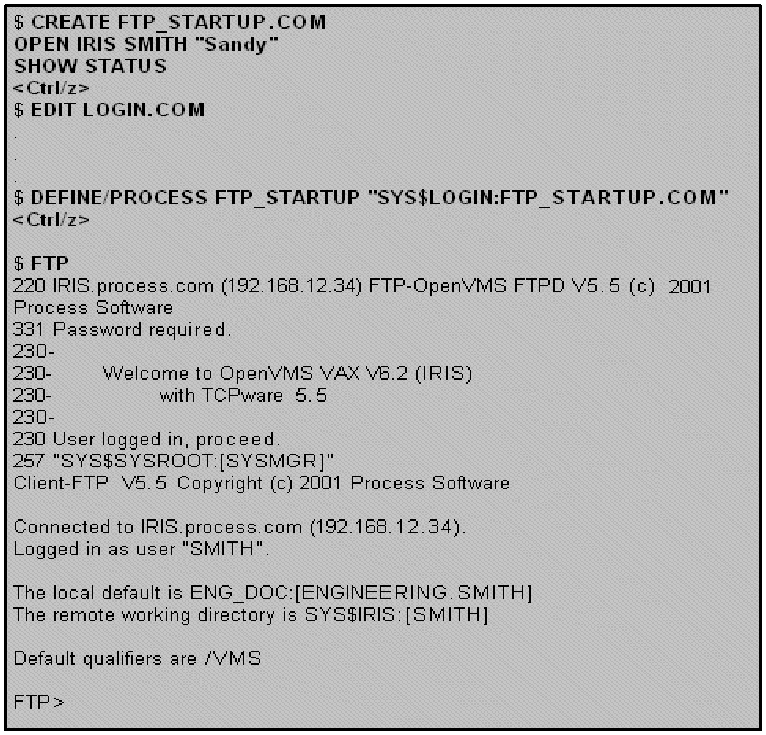
Note! FTP-OpenVMS does fast transfers between two OpenVMS systems using VMS file structure or VMS Plus Mode (for HP TCP/IP Services for OpenVMS (UCX) servers). When FTP-OpenVMS identifies file transfers between two OpenVMS hosts running TCPware, it automatically transfers files in large blocks rather than small records. These VMS modes greatly increase the transfer speed and preserve all Record Management Services (RMS) file attributes. The VMS modes are disabled with non-OpenVMS systems. See Table 3-2 for the file transfer format descriptions.

Graphical user interface method. To transfer files:

From local to remote	Click one or more files on the "Local" part of the **`File Transfers`** screen (see Figure 3-3) and click **`Copy-->`**. To give the file a specific name on the remote host, enter a filename in the **`New Remote File/Dir Name`** field.
From remote to local	Click one or more files on the "Remote" part of the **`File Transfers`** screen (see Figure 3-4) and click **`<--Copy`**. To give the file a specific name on the local host, enter a filename in the **`New Local Name`** field.

See the following information on symbolically linked UNIX systems.

Figure 3-7 GET, PUT, and COPY Command Format



```
$ CREATE FTP_STARTUP.COM
OPEN IRIS SMITH "Sandy"
SHOW STATUS
<Ctrl/z>
$ EDIT LOGIN.COM
.
.
.
$ DEFINE/PROCESS FTP_STARTUP "SYS$LOGIN:FTP_STARTUP.COM"
<Ctrl/z>

$ FTP
220 IRIS.process.com (192.168.12.34) FTP-OpenVMS FTPD V5.5 (c) 2001
Process Software
331 Password required.
230-
230-      Welcome to OpenVMS VAX V6.2 (IRIS)
230-              with TCPware 5.5
230-
230 User logged in, proceed.
257 "SYS$SYSROOT:[SYSMGR]"
Client-FTP V5.5 Copyright (c) 2001 Process Software

Connected to IRIS.process.com (192.168.12.34).
Logged in as user "SMITH".

The local default is ENG_DOC:[ENGINEERING.SMITH]
The remote working directory is SYS$IRIS:[SMITH]

Default qualifiers are /VMS

FTP>
```

Symbolic links in UNIX systems. UNIX systems can have files or directories pointing to other files or directories, known as symbolic links. TCPware treats symbolic links as directories, which appear in the **`Remote`**

Directories field on the menu screens. Once you click and perform an operation on a symbolic link, the directory name disappears from the **Remote Directories** field and the file to which it points appears in the **Remote Files** field. You can then treat the file like a regular UNIX file.

Anonymous Users

You can access some remote resources as an ANONYMOUS user instead of with your usual username and password. This is especially useful for access to sites such as the U.S. Library of Congress (LOCIS.LOC.GOV) that allow anonymous user access to some of their files.

Anonymous access depends on your use of the /ANONYMOUS qualifier with the FTP commands that require a file or directory specification using the node name syntax.

You can access some remote resources as an ANONYMOUS user in one of the following ways (see Figure 3-8):

1 By default, use the node name file syntax (as described below) with any FTP command that requires a file or directory specification (such as COPY, DIRECTORY, RENAME, and SET DEFAULT). This file syntax sends the ANONYMOUS username and your e-mail address as a password.

Thus, the following file or directory specification: node::path

is equivalent to: node"ANONYMOUS your-email-address"::path

With OpenVMS Alpha V6.1 and later, and all OpenVMS I64 systems, node can be a domain name or IP address.

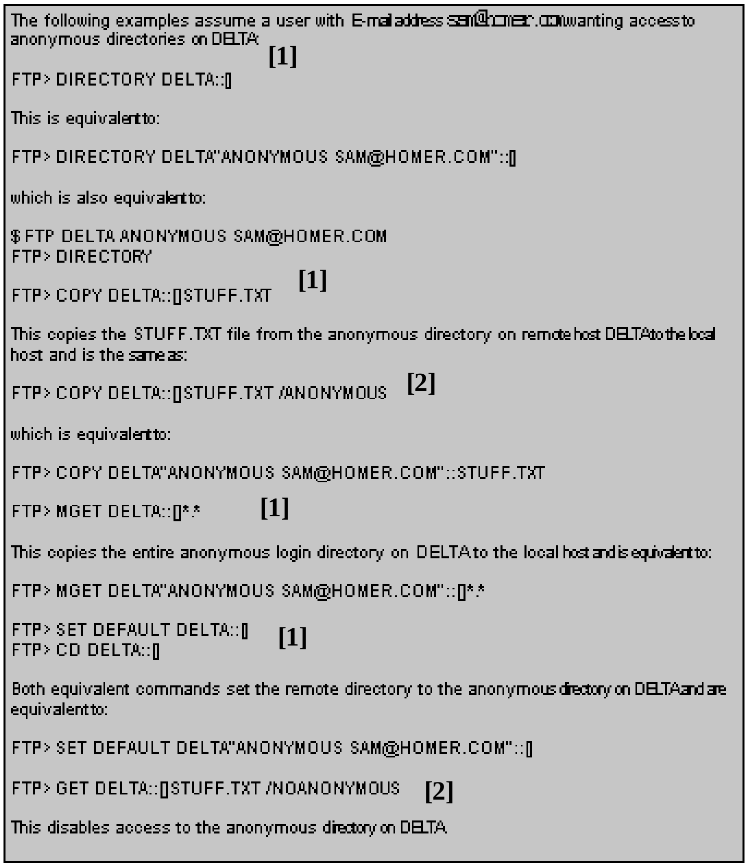
2 Use the filespec syntax described in Figure 3-8 and (optionally) add the /ANONYMOUS qualifier, or deny remote anonymous access using the /NOANONYMOUS qualifier.

Using the node name file syntax (and the /ANONYMOUS or /NOANONYMOUS qualifier) affects the following FTP commands:

COPY	CREATE/DIRECTORY	DELETE	DIRECTORY	DISPLAY
GET	LS	MDELETE	MGET	MKDIR
MPUT	PUT	RENAME	RMDIR	SET DEFAULT

Figure 3-8 shows examples of how to allow or deny anonymous user access to remote resources.

Figure 3-8 Anonymous User Access



```
The following examples assume a user with E-mail address sam@homer.com wanting access to
anonymous directories on DELTA:
                                          [1]
FTP> DIRECTORY DELTA::[]

This is equivalent to:

FTP> DIRECTORY DELTA"ANONYMOUS SAM@HOMER.COM"::[]

which is also equivalent to:

$ FTP DELTA ANONYMOUS SAM@HOMER.COM
FTP> DIRECTORY
                                          [1]
FTP> COPY DELTA::[]STUFF.TXT

This copies the STUFF.TXT file from the anonymous directory on remote host DELTA to the local
host and is the same as:

FTP> COPY DELTA::[]STUFF.TXT /ANONYMOUS   [2]

which is equivalent to:

FTP> COPY DELTA"ANONYMOUS SAM@HOMER.COM"::STUFF.TXT

FTP> MGET DELTA::[]*.*         [1]

This copies the entire anonymous login directory on DELTA to the local host and is equivalent to:

FTP> MGET DELTA"ANONYMOUS SAM@HOMER.COM"::[]*.*

FTP> SET DEFAULT DELTA::[]     [1]
FTP> CD DELTA::[]

Both equivalent commands set the remote directory to the anonymous directory on DELTA and are
equivalent to:

FTP> SET DEFAULT DELTA"ANONYMOUS SAM@HOMER.COM"::[]

FTP> GET DELTA::[]STUFF.TXT /NOANONYMOUS   [2]

This disables access to the anonymous directory on DELTA
```

Startup Command File

You can have a startup file execute FTP commands each time you invoke FTP. The startup file contains commands you want your system to perform at the beginning of each FTP session. Your system manager might already have defined a system-wide FTP startup file. Creating an FTP startup file is optional.

The startup command file in Figure 3-9 opens a remote connection, sends the password, and initiates a SHOW STATUS command.

You can set up an FTP startup command file or override one established by the system manager at the system level using the following procedure:

1 Create an FTP_STARTUP.COM file in your directory.

2 In the file, include the FTP commands you want executed each time you start an FTP session. If you include a password, make sure to use quotation marks to preserve case.

3 Edit your LOGIN.COM file and define the FTP_STARTUP logical to point to the startup file:

```
$ DEFINE/PROCESS FTP_STARTUP "SYS$LOGIN:FTP_STARTUP.COM"
```

Using the `DEFINE/PROCESS FTP_STARTUP` entry in the user's LOGIN.COM file causes that file to override any FTP startup command file at the system level.

4 Run FTP.

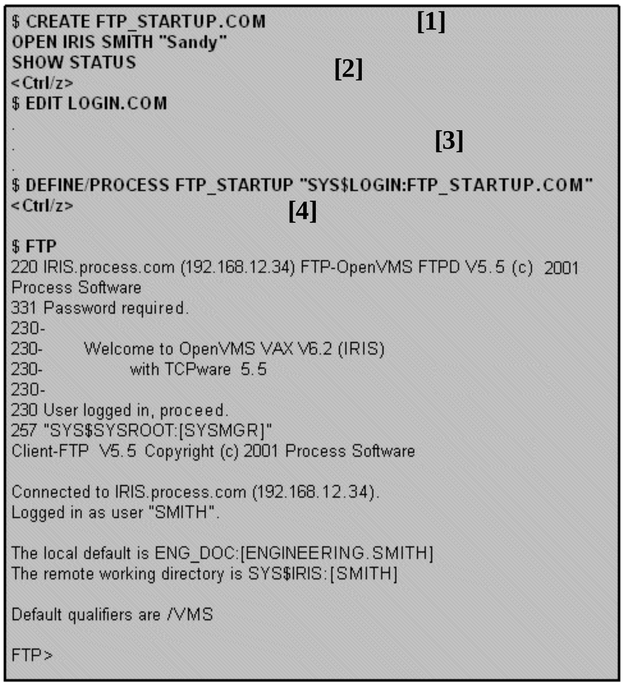
Whenever you run Client-FTP, it looks for the file to which the FTP_STARTUP logical points, and processes all the commands in that file.

If the EXIT or QUIT command appears in the startup file, Client-FTP:

– Ignores all commands following the EXIT or QUIT command.
– Continues with FTP operations after the startup command file.

Note! VERBOSE mode is set ON by default so that you can read replies from the FTP server when you connect or change server directories. This means that you do not need to include the SET DEBUG /CLASS=REPLIES (or its equivalent VERBOSE) command in the startup command file. Although an existing SET DEBUG /CLASS=REPLIES command in the file does not change the mode, a VERBOSE command toggles VERBOSE mode OFF. (See the SET DEBUG /CLASS command description in the *Command Reference*.) If you are an ANONYMOUS user, VERBOSE mode might help in reading any informational messages the FTP server creates.

Figure 3-9 Setting Up a Startup Command File



```
$ CREATE FTP_STARTUP.COM
OPEN IRIS SMITH "Sandy"
SHOW STATUS
<Ctrl/z>
$ EDIT LOGIN.COM
.
.
.
$ DEFINE/PROCESS FTP_STARTUP "SYS$LOGIN:FTP_STARTUP.COM"
<Ctrl/z>

$ FTP
220 IRIS.process.com (192.168.12.34) FTP-OpenVMS FTPD V5.5 (c)  2001
Process Software
331 Password required.
230-
230-       Welcome to OpenVMS VAX V6.2 (IRIS)
230-             with TCPware  5.5
230-
230 User logged in, proceed.
257 "SYS$SYSROOT:[SYSMGR]"
Client-FTP  V5.5 Copyright (c) 2001 Process Software

Connected to IRIS.process.com (192.168.12.34).
Logged in as user "SMITH".

The local default is ENG_DOC:[ENGINEERING.SMITH]
The remote working directory is SYS$IRIS:[SMITH]

Default qualifiers are /VMS

FTP>
```

Site-Specific Commands

The FTP-OpenVMS Server supports the SITE SPAWN and SITE SHOW TIME site-specific commands. The Client-FTP can issue these commands at any time.

Site-specific commands can vary depending on the remote FTP server; some servers do not support any.

Issue the FTP-OpenVMS site-specific commands in one of the following ways at the FTP> prompt (see Figure 3-10):

1 `SITE SHOW TIME`

This command returns the current date and the time of day for the OpenVMS system in the reply message.

2 `SITE SPAWN` *`dcl-command`*

This command allows you to execute any DCL command as a subprocess. You typically use this command to print files, submit batch jobs, execute command procedures, or issue other commands.

The screen does not display the output the subprocess generates. The system returns status from the subprocess as the status for the SITE SPAWN command.

Note! Spawning is not allowed for CAPTIVE accounts.

See the SITE and SPAWN commands in the *Command Reference*.

Figure 3-10 Issuing Site-Specific Commands

```
$ FTP
FTP> OPEN CONDOR
_Username [wombat]:WOMBAT
_Password:
FTP> SITE SHOW TIME
200 The date and time is "3-NOV-2001 11:3     [1]
FTP>

FTP> DIR
Directory DOC$DISK:[DOCUMENT.WOMBAT]

ANDY.TXT;1         4     4-NOV-2001    09:08:
CYN.PS;2          53    14-JAN-2001    14:10:
DNIP.TXT;1         8    10-JAN-2001    14:00:
DO_HELP.TXT;1      8    19-NOV-2001    09:49:

FTP> SITE SPAWN PRINT/QUE=ENG_PRINTER_ANSI ANDY.TXT     [2]
200 SITE command okay.
FTP>
```

Sample Session

This section describes a sample FTP-OpenVMS session.

See Figure 3-11 for the corresponding numbered steps. In this example, a user on local host BETA:

1 Starts Client-FTP, opens a connection to remote host THETA, and logs in as user SMITH (the display does not echo the password at the prompt). (If you are using Token Authentication, enter your PASSCODE in place of the password here.)

2 Using PUT, copies the local SYS.EXE file to THETA.

3 Using GET, copies the SYS.EXE file on THETA back to BETA.

4 Obtains a remote directory listing. There is a SYS.EXE file.

5 Deletes the SYS.EXE file.

6 Obtains another remote directory listing. SYS.EXE is now gone.

7 Obtains a local directory listing. Note that SYS.EXE;1 still exists locally.

8 Opens a connection to host ALPHA (running OpenVMS and FTP-OpenVMS) and logs in as USER. This closes the connection to THETA.

9 Obtains a remote directory listing on ALPHA.

10 Using GET, copies the ASCII file SCREEN_FTP.TXT on ALPHA to BETA.

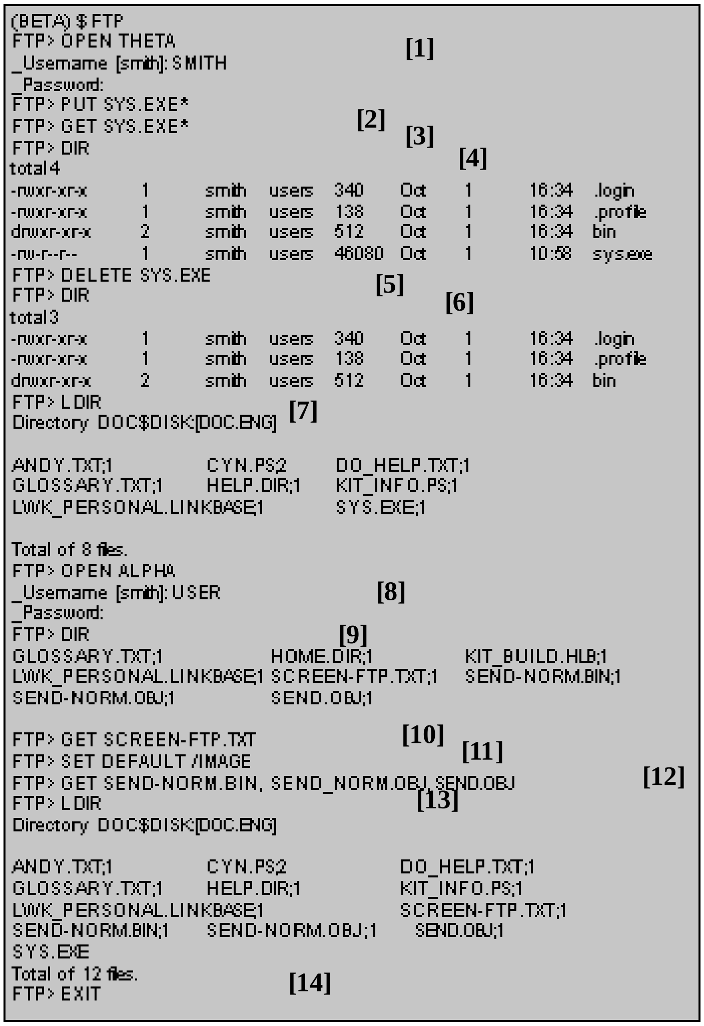
11 Changes the default for transferring files from formatted ASCII to IMAGE.

12 Using GET, copies the SEND-NORM.BIN, SEND-NORM.OBJ and SEND.OBJ files from ALPHA as image files on the local host.

13 Obtains a local directory listing. SCREEN-FTP.TXT, SEND-NORM.BIN, SEND-NORM.OBJ, and SEND.OBJ are now present.

14 Exits FTP.

Figure 3-11 Sample FTP-OpenVMS Session



```
(BETA) $ FTP
FTP> OPEN THETA
_Username [smith]: SMITH
_Password:
FTP> PUT SYS.EXE*
FTP> GET SYS.EXE*
FTP> DIR
total 4
-rwxr-xr-x    1    smith  users  340    Oct   1    16:34  .login
-rwxr-xr-x    1    smith  users  138    Oct   1    16:34  .profile
drwxr-xr-x    2    smith  users  512    Oct   1    16:34  bin
-rw-r--r--    1    smith  users  46080  Oct   1    10:58  sys.exe
FTP> DELETE SYS.EXE
FTP> DIR
total 3
-rwxr-xr-x    1    smith  users  340    Oct   1    16:34  .login
-rwxr-xr-x    1    smith  users  138    Oct   1    16:34  .profile
drwxr-xr-x    2    smith  users  512    Oct   1    16:34  bin
FTP> LDIR
Directory DOC$DISK:[DOC.ENG]

ANDY.TXT;1            CYN.PS;2     DO_HELP.TXT;1
GLOSSARY.TXT;1        HELP.DIR;1   KIT_INFO.PS;1
LWK_PERSONAL.LINKBASE;1            SYS.EXE;1

Total of 8 files.
FTP> OPEN ALPHA
_Username [smith]: USER
_Password:
FTP> DIR
GLOSSARY.TXT;1          HOME.DIR;1          KIT_BUILD.HLB;1
LWK_PERSONAL.LINKBASE;1 SCREEN-FTP.TXT;1    SEND-NORM.BIN;1
SEND-NORM.OBJ;1         SEND.OBJ;1

FTP> GET SCREEN-FTP.TXT
FTP> SET DEFAULT /IMAGE
FTP> GET SEND-NORM.BIN, SEND_NORM.OBJ, SEND.OBJ
FTP> LDIR
Directory DOC$DISK:[DOC.ENG]

ANDY.TXT;1            CYN.PS;2             DO_HELP.TXT;1
GLOSSARY.TXT;1        HELP.DIR;1           KIT_INFO.PS;1
LWK_PERSONAL.LINKBASE;1                    SCREEN-FTP.TXT;1
SEND-NORM.BIN;1       SEND-NORM.OBJ;1        SEND.OBJ;1
SYS.EXE
Total of 12 files.
FTP> EXIT
```

Command Reference

The following pages describe the FTP-OpenVMS commands. Table 3-3 contains command synonyms you can use interchangeably with FTP-OpenVMS commands. Table 3-4 shows commands you can use to do various tasks. Each command includes the graphical user interface equivalent, if available.

Enter FTP commands at the FTP> prompt. Client-FTP supports the following commands:

ACCOUNT	ENABLE VMS_PL	PWD	SET PASSIVE
CCC	ERROR_EXIT	QUOTE	SET VMS
CLOSE	EXIT	REMOTEHELP	SET STATUS
COPY	GET	RENAME	SITE
CREATE/DIR	HELP	SET BELL	SPAWN
DEFINE/KEY	LDIR	SET DEBUG	STRUCTURE
DELETE	OPEN	SET DEFAULT	TYPE
DIRECTORY	PROTECTION	SET HASH	USER
DISPLAY	PUT	SET LOWERCASE	

Table 3-3 FTP Command Synonyms

This command...	**Is a synonym for the FTP command...**
ASCII	TYPE ASCII
BELL	Toggles between SET BELL and SET NOBELL
BINARY or IMAGE	TYPE IMAGE
BYE or QUIT	EXIT
CD	SET DEFAULT /REMOTE
CONNECT	OPEN
DEBUG	Toggles SET DEBUG/CLASS=COMMANDS

DISCONNECT	CLOSE
H	HELP
HASH	Toggles between SET HASH and SETNOHASH
LCD	SET DEFAULT/LOCAL
LIST or LS	DIRECTORY/NAME_LIST
LOGIN	USER
MDELETE	DELETE/MULTIPLE
MGET	GET/MULTIPLE
MKDIR	CREATE/DIRECTCORY
MPUT	PUT/MULTIPLE
PASSIVE	Toggles between SET PASSIVE and SET NOPASSIVE
RECV	GET
RM	DELETE
RMDIR	DLETE/DIRECTORY
SEND	PUT
STATUS	HOW STATUS
VERBOSE	Toggles SET DEBUG/CLASS=REPLIES
Z	SPAWN

Table 3-4 Commands to Use to Perform Various Tasks on the Local System

DEFINE/KEY	Associate an equivalence string and set of attributes with a keyboard key
HELP	Bring up the Client-FTP online help facility

LCD	Set your local default directory
LDIR	List files in your local directory
SET BELL	Ring terminal bell after completing a file transfer
SET DEBUG	Display of debugging information
SET DEFLATE	Sets DEFLATE mode optional parameters. The only optional parameter currently recognized is /LEVEL, which can be specified as -1 (default, balance between compression and CPU time), 0 (no compression) to 9 (maximum compression). The SET MODE DEFLATE command must be used to enter deflate (ZLIB compression) mode. DEFLATE mode is not compatible with TLS authentication, which provides its own data compression algorithms. DEFLATE MODE cannot be used when TLS is being used.
SET HASH	Enable hash marks during a file transfer
SET LOWERCASE	Convert unquoted filenames to lowercase in a file transfer request
SET PASSIVE	Sets passive mode
SET VMS	FTP-Client negotiates with the server for VMS file structure when opening a connection
SHOW STATUS	Show the status of the current connection and local default directory
SPAWN	Executive DCL commands without exiting FTP
STRUCTURE	Change the default file structure for a transfer (FILE, RECORD, or VMS)
TYPE	Change the default file transfer format (ASCII, BINARY, IMAGE, FORTRAN, BLOCK, VARIABLE, or DEFAULT)

Table 3-5 Commands to Use to Perform various Tasks on the Remote System

CD	Change the remote default directory
DELETE	Delete a file or directory on the remote host

DIR, LIST, or ls	List files on the remote host
MKDIR	Create a directory on the remote host
PWD	Display the name of the current working directory on the remote host
QUOTE	Send an FTP command to the remote server
REMOTEHELP	Bring up the remote FTP server’s online help facility
RENAME	Rename a file on the remote host
SITE	Issue a site-specific command to the remote server
USER	Set the username at the remote host

Table 3-6 TCPware FTP Logicals for Users

FTP_STARTUP

Define the FTP_STARTUP logical to point to the FTP_STARTUP.COM file. For example:

```
$ DEFINE /SYSTEM /EXECUTIVE FTP_STARTUP SYS$MANAGER:FTP_STARTUP.COM
```

Client users can override this startup file by creating their own. Including the command DEFINE/PROCESS FTP_STARTUP in a user's LOGIN.COM file overrides any

TCPWARE_FTP_MAX_PRE_ALLOCATION

The logical TCPWARE_FTP_MAX_PRE_ALLOCATION may be defined to limit the size that a file will be pre-allocated to when file size information is available at transfer time. This can be important when transferring very large files, as it can take a long time to pre-allocate the file at the start of the transfer and timeout routines in FTP and/or firewalls may cause connections to be dropped. This logical does not have any effect for STRU OVMS transfers of Indexed, Contiguous, or Contiguous, Best Try files; these files need to have accurate allocation size information at the start of the transfer.

TCPWARE_ADD_CC_ON_FIXED_RECORD_FILES

If this logical is defined to TRUE and a file is transferred as TYPE IMAGE with QUOTE SITE RMS BLOCK OFF in effect, the FTP server will separate the records of a fixed length record file with the linefeed character. This is useful for avoiding the explicit conversion necessary when transferring the file to a non-VMS system with an FTP client that is not able to do record mode transfers.

TCPWARE_FTP_ALL_VERSIONS

Requests the NLST and LIST commands to display all versions of the specified files. If TCPWARE_FTP_ALL_VERSIONS is defined, the logical TCPWARE_FTP_STRIP_VERSION has no effect.

TCPWARE_FTP_ALL_VERSIONS is ignored if the FTP server is in UNIX emulation mode.

TCPWARE_FTP_DISALLOW_UNIX_STYLE

Controls whether UNIX style filename parsing is done. If not defined and a / is found in the filename, it is assumed to be a UNIX style filename.

```
$ DEFINE /SYSTEM /EXEC TCPWARE_FTP_DISALLOW_UNIX_STYLE FALSE
```

TCPWARE_FTP_EXTENSION_QUANTITY

Defines the default allocation/extension quantity for new files and appends. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /SYSTEM /EXEC TCPWARE_FTP_EXTENSION_QUANTITY n (number of
blocks)
```

TCPWARE_FTP_IGNORE_UNIX_DASH_OPTIONS

By default, the FTP server ignores Unix-style dash options on LIST and NLST when in Unix mode (for example, the “-l” in “ls -l”). Define this to be FALSE to tell the FTP server to pay attention to Unix-style dash options.

```
$ DEFINE /SYSTEM /EXEC TCPWARE_FTP_IGNORE_UNIX_DASH_OPTIONS FALSE
```

TCPWARE_FTP_KEEP_DIR_EXT

Sometimes the FTP server strips the .DIR extension from the file name of a directory when the NLST function is requested. The FTP server now looks for the logical TCPWARE_FTPD_KEEP_DIR_EXT and, if defined, does not remove the .DIR extension. To use this feature, define the logical as:

```
$ DEFINE TCPWARE_FTPD_KEEP_DIR_EXT TRUE
```

To return to the default behavior, deassign this logical.

TCPWARE_FTP_MESSAGE_FILE

Defines the message file the FTP user sees when connecting to the server or moving between directories. The definition of this logical is commented out but defined in the FTP_CONTROL.COM file as follows:

```
$ DEFINE TCPWARE_FTP_MESSAGE_FILE ".MESSAGE"
```

TCPWARE_FTP_NOKEEPALIVES

If this logical is defined, the FTP server will not send keepalives on the control channel. The KEEPALIVE command allows the FTP client program to toggle, whether or not it desires keepalives to be sent on the control channel. The SET [NO]KEEPALIVE command allows the FTP client to explicitly set whether or not it desires keepalives on the control channel.

TCPWARE_FTP_ONLY_BREAK_ON_CRLF

If this logical is set and an ASCII file is transferred, a new line is created in the file upon receipt of a carriage return/line feed sequence.

If this logical is not set and an ASCII file is transferred, a new line is created upon receipt of either a carriage return/line feed sequence or a line feed.

TCPWARE_FTP_SEMANTICS_FIXED_IGNORE_CC

If this logical is defined to TRUE, then GET operations of fixed lengths record files will not have a <CR>(carriage return)<LF>(line feed) added to the end of each record. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE TCPWARE_FTP_SEMANTICS_FIXED_IGNORE_CC ?
```

TCPWARE_FTP_SEND_FEAT_ON_CONNECT

By default, the FTP client sends the FEAT command upon connecting to a server. This can be disabled by defining this logical as FALSE.

```
$ DEFINE TCPWARE_FTP_SEND_FEAT_ON_CONNECT FALSE
```

When this is disabled the FTP client will not be able to detect the support of optional features such as TLS, REST STREAM, and others and these features may not work correctly if there is an attempt to use them.

TCPWARE_FTP_SERVER_LOG_LIMIT

By setting this logical in the LOGIN.COM file, you can specify that log files be retained. Set the logical name to a dash (-) to retain all log files, or specify a number in the range of 1 to 32000.

Directory size restrictions limit the number of potential files that can actually be created. If you do not specify a number or value, one log file is created or overwritten for each FTP session. Use the DCL PURGE command to delete unneeded log files. The following example specifies that 42 log files be retained:

```
$ DEFINE TCPWARE_FTP_SERVER_LOG_LIMIT 42
```

TCPWARE_FTP_STRIP_VERSION

Causes VMS mode output to have no versions. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /NOLOG TCPWARE_FTP_STRIP_VERSION ?
```

TCPWARE_FTP_USE_SRI_ENCODING_ON_ODS5

The logical TCPWARE_FTP_USE_SRI_ENCODING_ON_ODS5 can be defined to 1, TRUE or YES to cause the file name encoding used for UNIX-style file names on ODS-2 disks to be used on ODS-5 disks. This also sets the default case of letters in filenames to lowercase and ignores the stored case.

TCPWARE_FTP_UNIX_STYLE_BY_DEFAULT

If you define this logical, the FTP server starts in UNIX emulation mode.

The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /NOLOG TCPWARE_FTP_UNIX_STYLE_BY_DEFAULT ?
```

When sending the command from a non-OpenVMS client, a space is required between the file specification and the qualifier. For example:

```
$ GET filename /LOG
```

Previous command syntax: ftp>**put xx x.x/image=2048**
New command syntax: ftp>**put x.x "x.x/image=2048"**

You can disable this feature so that the FTP server can accept an OpenVMS transfer mode qualifier without including the space between the file specification and the qualifier. To disable this requirement, define the logical:

```
$ DEFINE TCPWARE_FTPD_NOUNIX_SYNTAX "TRUE"
```

TCPWARE_FTP_UNIX_STYLE_CASE_INSENSITIVE

Allows UNIX style filename handling to be case insensitive. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /NOLOG TCPWARE_FTP_UNIX_STYLE_CASE_INSENSITIVE ?
```

TCPWARE_FTPD_NOUNIX_SYNTAX

When sending a command to a non-OpenVMS client, a space is required between the file specification and the qualifier. For example:

FTP>**GET filename /LOG**

Previous command syntax: ftp>**put xx x.x/image=2048**
New command syntax: ftp>**put x.x "x.x/image=2048"**

You can disable this feature so that the FTP server can accept an OpenVMS transfer mode qualifier without including the space between the file specification and the qualifier. To disable this requirement, define the following logical:

```
$ DEFINE /SYSTEM /EXECUTIVE_MODE TCPWARE_FTPD_NOUNIX_SYNTAX "TRUE"
```

Troubleshooting

Access error messages help by entering `HELP TCPWARE MESSAGES [identifier]`, or connect to web site `http://www.process.com` (select `Customer Support` followed by the `Error Messages` button).

ACCOUNT

Specifies the user's account if the remote server requires it.

Format

ACCOUNT *account*

Parameter

account

User's account. Enclose in quotes if it contains special characters or embedded spaces, or contains mixed-case characters.

Example

The following specifies account Smith on the remote system. Use quotes around the mixed-case account name.

```
FTP>ACCOUNT "Smith"
```

CCC

Change the control port to clear text after performing RFC 4217 encrypted authentication. Clear text may be desired for the control port when NAT or firewalls are being used that expect to examine and/or alter commands and responses dealing with the data port (PORT, PASV, EPRT, EPSV and the respective replies). The PROTECTION command should be used before the CCC command as it is not allowed after the command channel has returned to clear text mode.

Format

CCC

Example

The following closes the current connection:

```
FTP>CCC
```

CLOSE

Closes the connection to the remote FTP server if one is open and keeps you in FTP.

OPEN and CONNECT also close an existing connection before opening another one.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ **`Connection`** ⇒ **`Close`**
or **`Open`**

Format

CLOSE

Synonym

DISCONNECT

Example

The following closes the current connection:

```
FTP>CLOSE
```

COPY

Copies files to or from a remote host. You specify whether the source or destination file is local or remote using the /LOCAL or /REMOTE qualifier. COPY supports full wildcard filespecs except wildcard symbols enclosed in a quoted string. Use the /MULTIPLE qualifier for a wildcard remote source filespec. /REMOTE also supports use of asterisk (*) wildcards after a semicolon (;) in remote file specifications. This creates the same version in the destination file as in the source file (instead of creating a new version). If the server is not OpenVMS, the version number is part of the filename. TCPware does not issue a warning if the server host already has a higher numbered version.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Select file(s) in **`Local Files`** for local-to-remote copy, or select file(s) in **`Remote Files`** for remote-to-local copy ⇒ **`Copy-->`** for a local-to-remote copy, or **`<--Copy`** for a remote-to-local copy. Give file new name, if desired, in **`New Local Name`** or **`New Remote File/Dir Name`**

Format

COPY *source [,source,...] destination*

Equivalents

GET= COPY *source* /REMOTE *destination*

RECV= COPY *source* /REMOTE *destination*

MGET= COPY *source* /REMOTE /MULTIPLE *destination*

PUT= COPY *source* /LOCAL *destination*

SEND= COPY *source* /LOCAL *destination*

MPUT= COPY *source* /LOCAL /MULTIPLE *destination*

Parameters

source

Input filespec. Use a comma between multiple filespecs. Enclose the filespec in quotes if you want to preserve case and did not use the SET NOLOWERCASE command. The format is:

`node"username password"::path`

node	hostname or DECnet node name (with OpenVMS Alpha V6.1 and later, and all OpenVMS I64 systems, the hostname can be a domain name or IP address)
username	valid account on the host
password	password (PASSCODE if using Token Authentication) for the account
path	location and name of the file

You can omit the node`"username password"::` part of the specification unless it is for a DECnet file. If omitted, Client-FTP uses the current default directory. You can use the *`node`*::*`path`* syntax (omitting the username and password) if you want access to anonymous FTP resources. In this case, FTP-OpenVMS implicitly adds the /ANONYMOUS qualifier.

Use the /LOCAL or /REMOTE qualifier after the parameter, depending on the context. The local filespec must conform to OpenVMS filenaming rules. The remote filespec must conform to the filenaming conventions of the remote host.

Enclose the *pathname* in quotes if it contains delimiters or symbols the FTP server could possibly misinterpret. For example, the following remote filespec is enclosed in quotes because it includes slashes (/) that OpenVMS normally interprets as qualifier delimiters:

```
ALPHA"smithabcd"::"/usr/bin/proj1.txt"
```

destination

Output filespec. Enclose the filespec in quotes if you want to preserve case and did not use the SET NOLOWERCASE command. If wildcarded (*), Client-FTP uses the source filename or extension, unless the filespec is a quoted string. See the source parameter for the destination filespec format.

To obtain the same version number in the destination file as in the source file (instead of creating a newer one), wildcard the destination file version using **;***. Note that if the server is not an OpenVMS host, the version number is included in the filename. You do not get a warning if the server host already has a higher numbered version. Also, if the server host already has the version specified, the old file with that version is overwritten.

Transfer Qualifiers (Positional)

/LOCAL

The preceding file is on the local host. If /LOCAL follows source, /REMOTE is implicit for *destination*. If /LOCAL is omitted, Client-FTP searches for a node; if found, Client-FTP assumes the file is remote. Do not use for both source and *destination*.

/REMOTE

The preceding file is on the remote host. If /REMOTE follows *source*, /LOCAL is implicit for *destination*. If /REMOTE is omitted, Client-FTP searches for a *node*; if found, Client-FTP assumes the file is remote. Do not use for both *source* and *destination*. (See the destination parameter on how to preserve version numbers on a remote copy.)

/MULTIPLE

Transfers multiple files. Use after *source* only. Include wildcards in *source* only because some remote hosts do not recognize the OpenVMS asterisk and percent characters as wildcards. The remote host's server must support the FTP NLST command. Not all servers support VMS files. If the server does and you do not specify another mode (using a qualifier or the STRUCTURE or SET DEFAULT commands), /VMS is the default.

File Type Qualifiers (Positional)

If you omit one of the file type qualifiers, Client-FTP transfers the file based on either:

- The current default setting; for example, ASCII or IMAGE.
- The extension (type) of the file you want to copy (see Client-FTP File Transfer Formats).

Setting a file type qualifier overrides the default transfer format for this transaction only. (See also theSET DEFAULT command.)

/ASCII

Transfers the preceding file in formatted ASCII format (see Client-FTP File Transfer Formats).

/BINARY

Transfers the preceding .BIN, .LDA, .OBJ, or .STB file in formatted binary format.

/BLOCK

Transfers the preceding STREAM, STREAM_CR, STREAM_LF, or UNDEFINED file in block mode (see Client-FTP File Transfer Formats).

/FORTRAN

Transfers the preceding file in FORTRAN mode. The first character of each record is a FORTRAN carriage control character. Some hosts do not recognize this transfer format.

/IMAGE*[=size]*

Transfers the preceding file in image mode. Optional *size* sets the record size of the local output file (see Client-FTP File Transfer Formats). Does not apply to remote output files. The maximum size for this qualifier is 32768.

/RECORD

Transfers the preceding file using STRU R so as to communicate the record structure during the copy. Not all servers support record structure mode. If you specify both /RECORD and /VMS, Client-FTP uses /VMS.

/VARIABLE

Transfers an image file (see /**IMAGE**) in variable length record mode. At the destination site, all /IMAGE records have a fixed length. Applies to local output image files only. This qualifier has meaning only if the /IMAGE qualifier is present.

/VMS

Transfers the preceding file in VMS file mode (see Client-FTP File Transfer Formats). Allows you to transfer any type of RMS file between OpenVMS systems. If you use /VMS, Client-FTP ignores /APPEND, /ASCII, /BINARY, /BLOCK, /FORTRAN, /IMAGE, and /VARIABLE. If you specify both /RECORD and /VMS, Client-FTP uses /VMS.

Other Qualifiers (Non-positional)

/ANONYMOUS
/NOANONYMOUS

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) anonymous user access to remote resources. You can omit /ANONYMOUS if you use the node file syntax (`node::pathname`). (See Anonymous Users.)

/APPEND

Appends the *source* file to the *destination* file. If the *destination* file does not exist, Client-FTP creates it. Only valid if appending to a file with the same file transfer type. Some remote hosts might not support this operation.

/CONFIRM
/NOCONFIRM (default)

/CONFIRM issues a confirmation prompt before copying a file. Useful when *source* contains wildcards so that you can confirm each file copy. Respond with **`Y`** or **`N`**. /NOCONFIRM is the default.

If confirming multiple file copying, use with COPY/MULTIPLE with a wildcard value. Position the qualifier immediately after the COPY verb to relate to all files, or after the particular filename to relate to that file only.

/CONTIGUOUS=*blocks*

Local output file should have an initial contiguous allocation of the specified number of *blocks*. If the output file is smaller than the specified *blocks*, Client-FTP truncates the allocation. If the output file is larger, the additional allocations are non-contiguous. Does not apply to remote output files.

/FDL

Uses and then deletes a separate FDL file describing the specified file's OpenVMS RMS record attributes. This qualifier is useful after a PUT /FDL operation from a VMS node transfers a file to a non-VMS node: the GET /FDL operation can then return the file with the proper record attributes back from the non-VMS node. The default is not to create an accompanying FDL file. The TYPE command determines the type of file. A transfer of:

- ASCII data results in a sequential file with variable length records (the default).
- IMAGE data results in a sequential file with fixed length records of 512 bytes.

/IGNORE
/NOIGNORE (default)

/IGNORE ignores errors so that copying can continue with the next file. /NOIGNORE terminates copying if an error occurs.

/LOG
/NOLOG (default)

/LOG displays file specifications for each file transferred. /NOLOG does not display the transferred file's specifications.

/RESTART

For STREAM mode transfers restart the transfer where it was interrupted. The client verifies that the server supports the RFC 3659 SIZE and REST commands, and ignores the qualifier if it does not.

This does NOT work for VMS mode transfers (STRU VMS), and if the remote system is a VMS system it is recommended that a STRU FILE be done before the transfer command and to include /NOVMS on the command line.

/SET_FACTS

Set selected file facts on the destination file to match the source file after transfer. The facts currently supported are:

- MODIFICATION__TIME

Examples

1 Each of these commands copies the STUFF.TXT file from the local host to remote host SYS1 (the receiving system stores the file under the same filename in user SMITH's directory):

```
FTP>COPY STUFF.TXT SYS1"SMITH SECRET"::
FTP>PUT STUFF.TXT SYS1"SMITH SECRET"::
```

2 Each of these commands copies the DATA1.TXT and DATA2.TXT files from the remote host to the local host, assuming that a connection to the remote host is currently open:

```
FTP>COPY DATA1.TXT,DATA2.TXT /REMOTE *
FTP>GET DATA1.TXT,DATA2.TXT
```

3 Each of the following commands copies all .BAS files from a remote OpenVMS host to the local host. The /MULTIPLE qualifier and the asterisk wildcard are used in the COPY command, and they are omitted in the equivalent MGET command.

```
FTP>COPY *.BAS/REMOTE/MULTIPLE *
FTP>MGET *.BAS
```

4 The issuer of the following command wants to copy all local .SQL type files into multiple files in the remote UNIX system's directory.

```
FTP>COPY *.SQL/LOCAL/MULTIPLE "/usr/users/sql/*"
```

To accomplish this, the issuer uses an asterisk wildcard in the output filespec, as in Example 3. However, the result is not as intended. Because the asterisk is part of a quoted string, the command actually copies the files into a single file literally named * on the remote host.

To avoid this, set the remote default directory to the full pathname. You do not have to specify the quoted pathname in the COPY command:

```
FTP>SET DEFAULT/REMOTE "/usr/users/sql"
FTP>COPY *.SQL/LOCAL/MULTIPLE *
```

The asterisk now acts as a true wildcard, with the intended result.

CREATE/DIRECTORY

Creates a directory on the remote host. The /DIRECTORY qualifier is required as part of the command. Some remote hosts might not support directory creation operations.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Enter new directory name in

`New Remote File/Dir Name` ⇒ **`Make Dir>`**

Format

CREATE/DIRECTORY *remote-directory*

Synonym

MKDIR

Parameter

remote-directory

Directory to create on the remote host, in the format:

```
[node"username password"::]directory
```

To open a connection first, use the *`node"username password"`*`::`part of the format. This syntax is optional. If you omit the parameter and a connection is already open, Client-FTP uses the current default directory. The *directory* part of the format is any valid remote directory specification. Enclose the specification in quotes if it contains special characters or embedded spaces, or is case-sensitive.

Use the *`node`*::*`directory`* syntax to create an anonymous user directory. The /ANONYMOUS qualifier is implicit.

Qualifier

/**ANONYMOUS**
/**NOANONYMOUS**

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) creation of anonymous user directories. You can omit /ANONYMOUS if using the node file syntax (*`node`*::*`pathname`*). (See Anonymous Users.)

Examples

1 These commands are equivalent and create a directory USERS on the remote OpenVMS host SYS1, with the username and password specified explicitly:

```
FTP>CREATE/DIRECTORY SYS1"SMITH SECRET"::[USERS]
FTP>mkdir sys1"smith secret"::[users]
```

2 All three of the following commands create a directory USERS in the anonymous directory on the remote OpenVMS host SYS2.

```
FTP>CREATE/DIRECTORY SYS2::[USERS]
FTP>mkdir sys2::[users]
FTP>mkdir sys2::[users] /anonymous
```

The commands are equivalent to:
FTP>**CREATE/DIRECTORY SYS2"ANONYMOUS *user-email-address*"::[USERS]**

DEFINE/KEY

Associates an equivalence string and a set of attributes with a key on the terminal keyboard.

Format

DEFINE/KEY *key-name ["]equivalence-string["]*

Parameters

key-name

Name of the key to define. Table 3-7 lists key designations for three terminal types:

- On LK201 terminals, you can define three types of keys: numeric keypad, editing keypad (except the up and down arrow keys), and function key row (except F1 through F5).
- On VT100-type terminals, you can also define the left arrow and right arrow keys. On VT200 terminals, the left arrow and right arrow keys, and the F6 through F14 keys, are for command line editing. Issue the DCL command SET TERMINAL/ NOLINE_EDITING to define these keys before you run Client-FTP. You can also press Ctrl/V to enable keys F7 through F14 (but not F6).
- On VT52 terminals, the only definable keys are on the numeric keypad.

Table 3-7 Key Designations for Three Terminal Types

Key Name	**LK201**	**VT100-type**	**VT52**
PF1	PF1	PF1	[blue]
PF2	PF2	PF2	[red]
PF3	PF3	PF3	[gray]
PF4	PF4	PF4	
KP0,...,KP9	0,...,9	0,...9	0,...9
PERIOD	.	.	.
COMMA	,	,	,
MINUS	-	-	-
ENTER	ENTER	ENTER	ENTER
LEFT	←	←	←
RIGHT	→	→	→

Find (E1)	Find		
Insert Here (E2)	Insert_Here		
Remove (E3)	Remove		
Select (E4)	Select		
Prev Screen (E5)	Prev_Screen		
Next Screen (E6)	Next_Screen		
HELP	Help		
DO	Do		
F6,...,F20	F6,...,F20		

equivalence-string

String to substitute when you press the key. If the string contains spaces, enclose it in quotes.

Qualifiers

/ECHO
/NOECHO (default)

/ECHO displays the equivalence string on your screen after you press the key. /NOECHO is the default. Do not use /NOECHO with /NOTERMINATE.

/IF_STATE=*(state-name,...)*
/NOIF_STATE (default)

/IF_STATE specifies a list of one or more *state-names* (an alphanumeric string) for the key definition to be in effect. If you specify only one *state-name*, you can omit the parentheses. By including several *state-names*, you can define a key to have the same function in all the specified states. /NOIF_STATE is the default, where Client-FTP uses the current state.

Establish states using /SET_STATE.

/LOCK_STATE
/NOLOCK_STATE (default)

/LOCK_STATE specifies that the state set by /SET_STATE remains in effect until explicitly changed. /NOLOCK_STATE is the default, meaning the state which has been set in effect by /SET_STATE is in effect only for the next definable key you press or the next read-terminating character you type.

You can specify /LOCK_STATE only on the same command line as /SET_STATE.

/**SET_STATE=*state-name***
/NOSET_STATE (default)

/SET_STATE specifies the *state-name* (an alphanumeric string) you want set for the key. The default is /NOSET_STATE, where the current state locked by /LOCK_STATE is in effect.

/**TERMINATE**
/NOTERMINATE (default)

/TERMINATE specifies that Client-FTP terminates (effectively executes) the current equivalence string when someone presses the defined key. /NOTERMINATE allows you to create key definitions that insert text into command lines, after prompts, or into other typed text.

Example

The following sets the F1 key on the keyboard to the ""SMITH SECRET"::[USERS]" string, sets the state to 1, and locks the state for that definition:

```
FTP>DEFINE/KEY F1 """SMITH SECRET""::[USERS]" /SET=1 /LOCK
```

DELETE

Deletes files or directories on the remote host.

Some remote hosts might not support file or directory deletion operations.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Select file (or files) in **`Remote Files`** or directory (or directories) in **`Remote Directories`** ⇒ **`Delete->`** for files or **`Del Dir>`** for directories.

The remote file listing displays version numbers of files. Only the highest-numbered version appears in the list. Use **`Refresh>`** to refresh the remote listing display.

Format

DELETE ***file[,file,...]***

Synonyms

RMDIR ***dir[,dir,...]*** = DELETE /DIRECTORY

MDELETE ***file[,file,...]*** = DELETE /MULTIPLE

CAUTION! The DIRECTORY command does not list hidden files (files that start with a period). Using any wildcards with the MDELETE command deletes hidden files, which you might need.

Parameters

file

dir

Remote files or directories to delete. If used with the /DIRECTORY qualifier, you can indicate the remote directory in the format:

`[`***`node"username password"`***`::]`***`directory`***

To open a connection first, use the `node"username password"`:: part of the format. This syntax is optional. If you omit the parameter and a connection is already open, Client-FTP uses the current default directory. The *directory* part of the format is any valid remote directory specification. Enclose the specification in quotes if it contains special characters or embedded spaces, or is case-sensitive.

Use the *`node::directory`* syntax for access to an anonymous user directory. The /ANONYMOUS qualifier is implicit.

When deleting files, *file* can contain wildcards. See the /MULTIPLE qualifier.

Qualifiers

/**ANONYMOUS**
/**NOANONYMOUS**

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) deletion of anonymous files or directories. You can omit /ANONYMOUS if using the node file syntax (*`node::path`*). (See Anonymous Users.)

Note! SET DEFAULT can change the defaults indicated for the following qualifiers.

/**CONFIRM**
/**NOCONFIRM** (default)

/CONFIRM issues a confirmation prompt before deleting a file. Useful when source contains wildcards so that you can confirm each file copy. Respond with **Y** or **N**. /NOCONFIRM is the default.

If confirming multiple file deletions, use with MDELETE or DELETE/MULTIPLE with a wildcard value. Position the qualifier immediately after the DELETE verb to relate to all files, or after the particular filename to relate to that file only.

/**DIRECTORY**

Deletes a directory (equivalent to **RMDIR**). If omitted, Client-FTP deletes a file. Do not use with /MULTIPLE.

/**IGNORE**
/**NOIGNORE** (default)

/IGNORE ignores errors so that deletion can continue with the next file when using /MULTIPLE. /NOIGNORE terminates the deletion operation if an error occurs.

/**LOG**
/**NOLOG** (default)

/LOG displays file specifications for each file deleted.

/**MULTIPLE**

Deletes multiple files (equivalent to **MDELETE**). You must include wildcards in the filespec. /MULTIPLE is necessary because other systems do not universally recognize the OpenVMS asterisk and percent characters as wildcards. (You do not need this qualifier with multiple deletes between OpenVMS systems.) The remote host's FTP server must support the FTP NLST command for remote wildcard operations to work. Do not use with /DIRECTORY.

Examples

1 The following deletes the `proj1` file from the UNIX `/usr/src/`directory:

```
FTP>DELETE "/usr/src/proj1"
```

2 The following deletes all files with the .TMP extension in the remote default directory. You do not need /MULTIPLE when doing this delete operation between OpenVMS systems. If several versions of any *.TMP file exist, it deletes only the latest version.

```
FTP>DELETE *.TMP/MULTIPLE
```

3 The following deletes all files with the FOO filename in the remote default directory. You do not need /MULTIPLE when doing this delete operation between OpenVMS systems. If several versions of any FOO.* file exist, it deletes only the latest version.

```
FTP>DELETE FOO.*/MULTIPLE
```

4 The following deletes all files and file versions with the FOO filename in the remote default directory. For example, this command deletes FOO.EXE;1, FOO.EXE;2, FOO.C;1, FOO.C;2, and FOO.TXT;1. You do not need /MULTIPLE when doing this delete operation between OpenVMS systems.

```
FTP>DELETE FOO.*;*/MULTIPLE
```

DIRECTORY

Lists files on the remote host. If the remote host is a TCPware host, also lists the creation date and file type.

See LDIR to list files on the local host.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Enter directory path in **`Current Remote Directory`** ⇒ **`Refresh>`**

Format

DIRECTORY *[directory]*

Synonym

LS *[directory]*= DIRECTORY {/BRIEF | /NAME_LIST}

Parameter

directory

Directory to list on the remote host, in the format:

```
[node"username password"::]directory
```

To open a connection, use the `node"username password"::` part of the format. This syntax is optional. If you omit the parameter and a connection is open, Client-FTP uses the current default directory. The *directory* part of the format is any valid remote directory specification. Enclose the specification in quotes if it contains special characters or embedded spaces, or is case-sensitive.

Use the `node::directory` syntax for access to an anonymous user directory.
The /ANONYMOUS qualifier is implicit.

Qualifiers

/ANONYMOUS
/NOANONYMOUS

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) anonymous user access to remote resources. You can omit /ANONYMOUS if using the directory syntax `node::directory`. (See Anonymous Users.)

/BRIEF

/NAME_LIST

Returns a list of filenames instead of a normal directory listing (equivalent to **LS**). Uses the FTP NLST command. /BRIEF and /NAME_LIST are synonyms.

/OUTPUT=*file*

Filespec for a local file to receive the directory listing. If omitted, the directory is displayed on your terminal.

Examples

1 The following returns a listing for the remote /usr/src/ UNIX directory, assuming that a connection to the remote host is open:

```
FTP>DIRECTORY "/usr/src/"
```

2 The following returns a listing for the remote SYS$SYSTEM directory, assuming that a connection to the remote host is open:

```
FTP>DIRECTORY SYS$SYSTEM:
```

DISPLAY

Displays a remote file on the screen.

Equivalent to the GET (or COPY /REMOTE) command with SYS$OUTPUT as the local file specification.

If a VMS Plus mode transfer is requested, DISPLAY temporarily cancels VMS Plus mode, transfers the file(s), and resets VMS Plus mode again.

Note that displaying a non-ASCII file might produce unrecognizable output, as would be the case with the DCL TYPE command.

Format

DISPLAY ***remote-file[,remote-file,...]***

Equivalents

COPY ***remote-file[,remote-file,...]*** **/REMOTE [/MULTIPLE] SYS$OUTPUT**

[*M*]*GET** ***remote-file[,remote-file,...] **SYS$OUTPUT**

Parameters

remote-file

Input filespec on the remote host. Enclose in quotes if you want to preserve case and did not use the SET NOLOWERCASE command, or the filespec contains delimiters or symbols the FTP server can interpret in special ways. Use a comma between multiple filespecs. The remote filespec must conform to the filenaming conventions of the remote host.

Examples

The following shows formats of acceptable equivalent commands that implement the DISPLAY function:

```
FTP>DISPLAY TEXT.TXT
FTP>GET TEXT.TXT SYS$OUTPUT
FTP>MGET TEXT.TXT, TEXT2.TXT SYS$OUTPUT
FTP>COPY TEXT.TXT /REMOTE SYS$OUTPUT
FTP>COPY TEXT.* /REMOTE /MULTIPLE SYS$OUTPUT
FTP>COPY NODE"USER PASSWORD"::TEXT.TXT SYS$OUTPUT
```

ENABLE *[*DISABLE*]* VMS_PLUS

Turns VMS Plus Mode on or off. This lets you specify a transfer mode based on file type, for example, ASCII or image.

In VMS Plus mode, file transfers use File Descriptor Language (FDL) information to create output files.

Format

ENABLE VMS_PLUS

DISABLE VMS_PLUS

ERROR_EXIT

Exits FTP with a specified status if an error occurs in the previous FTP command. This feature is useful when running FTP from a command procedure.

Note that you exit FTP-OpenVMS if you try to use this command interactively.

Format

ERROR_EXIT *[status]*

Parameter

status

Optional status value the DCL $STATUS symbol returns if FTP exits. Specifies which command (or sequence of commands) failed. If omitted, Client-FTP uses the status value of the last error.

Note! Client-FTP reports the $STATUS as the status value ORd with `%X10000000`.

Example

The following example is part of a DCL command procedure:

```
.
$ SET NOON
$ FTP
OPEN LILAC SMITH PASSWORD
ERROR_EXIT %X10000010
PUT DATA_FILE1.TXT
ERROR_EXIT %X10000020
PUT DATA_FILE1.IMG
ERROR_EXIT %X10000030
PUT DATA_FILE1.DES
ERROR_EXIT %X10000040
EXIT
$ FTP_EXIT_STATUS = $STATUS
$ SET ON
$ IF (FTP_EXIT_STATUS .EQ. %X10000010) THEN GOTO LOGIN_FAILED
$ IF (FTP_EXIT_STATUS .EQ. %X10000020) THEN GOTO TRANSFER_1_FAILED
.
.
```

This command procedure transfers several files and uses ERROR_EXIT to detect if any of the transfers fail. FTP_EXIT_STATUS returns the following values:

- `%X10000010` if the connection or login to LILAC fails
- `%X10000020` if FTP cannot transfer DATA_FILE1.TXT
- and so on
- `1` if the connection is successful

EXIT

Exits FTP and returns to the DCL prompt.

If a connection is open, Client-FTP closes it before exiting.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ `File` ⇒ `Exit`

Format

EXIT

Synonyms

QUIT

BYE

GET

Copies files from a remote host.

GET supports full wildcard filespecs except wildcards enclosed in a quoted string. Use the /MULTIPLE qualifier for a wildcarded remote filespec.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Select file (or files) in **`Remote Files`**

`Remote Files` ⇒ **`<--Copy`**. Give file new name, if desired, in **`New Local Name`**

Format

GET *remote-file[,remote-file,...] [local-filename]*

Equivalents

COPY *remote-file* /REMOTE *local-filename*

MGET *wildcarded-remote-files* = GET *remote-file* /MULTIPLE

RECV *remote-file[,remote-file,...] [local-filename]*

Parameters

remote-file

Input filespec on the remote host. Enclose in quotes if you want to preserve case and did not use the SET NOLOWERCASE command, or the filespec contains delimiters or symbols the FTP server can interpret in special ways. Use a comma between multiple filespecs.

The remote filespec must conform to the filenaming conventions of the remote host. In OpenVMS-to-OpenVMS file transfers, the *remote-file* and *local-filename* formats are the same. (See the *local-filename* parameter).

wildcarded-remote-files

Input filespec on the remote host in wildcarded format. Wildcards include the `%` or `?` symbol to indicate individual characters, and the `*` symbol to indicate multiple characters. Examples of wildcarded filespecs are `*.txt` , `W????.*`, and `*.*;*`.

local-filename

Output filespec on the local host. If omitted, Client-FTP uses the *remote-file* filename (and extension if it exists), unless *remote-file* is a quoted string. If used, must conform to the OpenVMS filenaming format:

`node"username password"::path`

node	hostname or DECnet node name (with OpenVMS Alpha V6.1 and later, and all OpenVMS I64 systems, the host name can be a domain name or IP address)
username	valid account on the host
password	password (PASSCODE if using Token Authentication) for the account
path	location and name of the file

You can omit the node `"username password"`:: part of the specification unless it is for a DECnet file. If omitted, Client-FTP uses the current default directory.

You can use the *`node`*`::`*`path`* syntax (omitting the username and password) if you want access to anonymous FTP resources, in which case the /ANONYMOUS qualifier is implied.

Qualifiers

If you omit one of the file type qualifiers (/ASCII, /BINARY, /FORTRAN, /IMAGE, /VMS), Client-FTP transfers the file based on either:

- The current default setting; for example, ASCII or IMAGE.
- The extension (type) of the file you want copied (see Client-FTP File Transfer Formats).

Setting a file type qualifier overrides the default transfer format for this transaction only. See also the SET DEFAULT command.

/**ANONYMOUS**
/**NOANONYMOUS**

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) anonymous user access to remote resources. You can omit /ANONYMOUS if using the node file syntax (*`node`*`::`*`path`*). (See Anonymous Users.)

/**APPEND**

Appends the *remote-file* file to the *local-filename*. If the *local-filename* does not exist, Client-FTP creates it. Some remote hosts do not support this operation. **NOTE:** If the operation fails, try appending in binary mode by using the /BINARY qualifier.

/**ASCII**

Transfers the file in formatted ASCII format (see Client-FTP File Transfer Formats).

/**BINARY**

Transfers .BIN, .LDA, .OBJ, and .STB, files in formatted binary format (see Client-FTP File Transfer Formats).

/**BLOCK**

Transfers STREAM, STREAM_CR, STREAM_LF, and UNDEFINED files in block mode (see Client-FTP File Transfer Formats).

/**CONFIRM**
/**NOCONFIRM** (default)

/CONFIRM issues a confirmation prompt before getting a file. Useful when source contains wildcards so that you can confirm each file copy. Respond with **`Y`** or **`N`**. /NOCONFIRM is the default.

If confirming multiple file gets, use with MGET or GET/MULTIPLE with a wildcard value. Position the qualifier immediately after the GET verb to relate to all files, or after the particular filename to relate to that file only.

/**CONTIGUOUS=blocks**

Local output file should have an initial contiguous allocation of the specified number of *blocks*. If the output file is smaller than the specified *blocks*, Client-FTP truncates the number of blocks allocated. If the output file is larger, the additional allocations are non-contiguous. Does not apply to remote output files.

/FDL

Uses and then deletes a separate FDL file describing the specified file's OpenVMS RMS record attributes. This qualifier is useful after a PUT /FDL operation from a VMS node transfers a file to a non-VMS node: the GET /FDL operation can then return the file with the proper record attributes back from the non-VMS node. The default is not to create an accompanying FDL file. The TYPE command determines the type of file. A transfer of:

- ASCII data results in a sequential file with variable length records (the default).
- IMAGE data results in a sequential file with fixed length records of 512 bytes.

/FORTRAN

Transfers the file in FORTRAN mode (see Client-FTP File Transfer Formats). The first character of each record is a FORTRAN carriage control character. Some hosts do not recognize this transfer format.

/IGNORE
/NOIGNORE (default)

/IGNORE ignores errors so that copying can continue with the next file. /NOIGNORE terminates copying if an error occurs.

/IMAGE*[=size]*

Transfers the file in image mode. Optional size sets the record size of the local output file (see Client-FTP File Transfer Formats). Does not apply to remote output files.

/LOG
/NOLOG (default)

/LOG displays file specifications for each file transferred.

/MULTIPLE

Transfers multiple files (equivalent to **MGET**). Use after *remote-file* only and include wildcards in *remote-file*. Necessary because some remote hosts do not recognize the OpenVMS asterisk, percent, or question mark characters as wildcards. /MULTIPLE ensures that the remote host understands more than one file is to be transferred. The remote host's server must support the FTP NLST command for remote wildcard operations to work.

/RECORD

Transfers the preceding file using STRU R so as to communicate the record structure during the copy. A positional qualifier. Not all servers support record structure mode. If you specify both /RECORD and /VMS, Client-FTP uses /VMS.

/RESTART

For STREAM mode transfers restart the transfer where it was interrupted. The client verifies that the server supports the RFC 3659 SIZE and REST commands, and ignores the qualifier if it does not.

This does NOT work for VMS mode transfers (STRU VMS), and if the remote system is a VMS system it is recommended that a STRU FILE be done before the transfer command and to include /NOVMS on the command line.

/SET_FACTS

Set selected file facts on the destination file to match the source file after transfer. The facts currently supported are:

MODIFICATION__TIME

/VARIABLE

Transfers an image file (see /IMAGE) in variable length record mode. All /IMAGE records are fixed length when stored at the destination. Applies to local output image files only.

/VMS

Transfers the file in VMS file mode (see Client-FTP File Transfer Formats). Allows you to transfer any type of RMS file between OpenVMS systems. A positional qualifier. If you use /VMS, Client-FTP ignores /APPEND, /ASCII, /BINARY, /BLOCK, /FORTRAN, /IMAGE, and /VARIABLE. If you specify both /RECORD and /VMS, Client-FTP uses /VMS.

Not all servers support VMS files. If the server does and you do not specify another mode (using a qualifier or the STRUCTURE or SET DEFAULT commands), /VMS is the default.

Examples

1 The following copies the DATA1.TXT and DATA2.TXT files from the remote host to the local system, assuming that a connection to the remote host is currently open:

```
FTP>GET DATA1.TXT,DATA2.TXT
```

2 The following copies all remote files with extension .BAS from a remote OpenVMS host to the local host:

```
FTP>MGET *.BAS
```

3 The following copies the STUFF.TXT file from DELTA's anonymous directory. It is equivalent to having used /ANONYMOUS. Sends the *"ANONYMOUS user-email-address"* username and password with the command.

```
FTP>RECV DELTA::STUFF.TXT
```

HELP

Accesses the Client-FTP online help.

Client-FTP help uses the OpenVMS interactive help facility.

To exit the help facility, press **`Return`** until you return to the `FTP>` prompt.

See the REMOTEHELP command, or the /REMOTE qualifier, for access to the remote server's online help.

Format

HELP *[/REMOTE] [topic]*

Synonyms and Equivalents

H

REMOTEHELP *[topic]* = HELP /REMOTE *[topic]*

HELP /REMOTE SITE = REMOTEHELP SITE = SITE HELP = QUOTE HELP SITE

Parameter

topic

Optional; allows you to specify the topic, if known, for which you want help. Otherwise HELP offers you a list of topics from which to choose.

Qualifier

/REMOTE

Equivalent to the REMOTEHELP command: it accesses the remote FTP server's online help instead of the local Client-FTP online help.

Position the qualifier directly after the HELP command. If positioned after the *topic*, you could get incorrect help or an error. For example, if you specify HELP LDIR /REMOTE, you get on-line help for "LDIR /REMOTE," which does not exist.

LDIR

Lists files in your local directory along with their creation date and size.

See DIRECTORY to list files on the remote host.

See SET DEFAULT /LOCAL to set the default local directory.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Enter directory path in **`Local Files`** ⇒ **`Filter`**

Format

LDIR ***[directory]***

Equivalent

SPAWN DIRECTORY ***[directory]***

Parameter

directory

Directory to list on your local host. The asterisk (*) wildcard is acceptable.

OPEN

Opens a connection to a remote host.

The connection remains open until you exit FTP, close the connection with the CLOSE command, or open a new connection using the OPEN command or any other command that accepts a node specification.

Graphical User Interface Equivalent

TCPware FTP-OpenVMS File Transfers ⇒ Connection ⇒ Open

Format

OPEN *[host [username [password [account]]]]*

If you:

- Supply the host, username, password, and account (if required) with the command, you are not prompted for them separately.
- Omit the parameters from the command line, you are prompted for them.
- Use the OPEN command non-interactively (for example, a batch job), and do not want to be prompted for a username, password, and account, then include the parameters on subsequent lines, after the OPEN command, in the command file.
- Want to be prompted for a password, do not submit the command file with a batch job.

The display does not echo the password or account information. After a connection is open, you do not have to specify the parameters for remote files.

Synonym

CONNECT

Parameters

host

Name or internet address of the remote host to which you want to connect. OPEN supports any valid hostname syntax, including an internet address.

username

Username on the remote host. Enclose the username in quotes if the case is important or it contains special characters. For a null username, use a pair of quotation marks (" ").

password

Password on the remote host. Enclose the password in quotes if the case is important or it contains special characters. For a null password, use a pair of quotation marks (" ").

If you use OPEN at the DCL level (see the second example), include the password on the same command line.

If you are designated by the system administrator as having password authentication using Token Authentication, you need to enter the PASSCODE in place of the password. Depending on which type of SecurID card you were assigned:

- Enter a combination of your personal identification number (PIN) and the tokencode that appears on the card (with no separating space) as the password, or
- Enter your PIN on the PINPAD] card and the resulting tokencode that appears on the card as the password.

See Chapter 15, *Token Authentication: Protecting Logins*, for details on obtaining PASSCODEs. account

Account on the remote host. Enclose the account in quotes if the case is important or it contains special characters.

Qualifiers

/**PORT**=*port*

Port number for the remote FTP server. If omitted, Client-FTP uses port number 21.

/**TIMEOUT**=*time*

Timeout time, in seconds, to establish the FTP control connection. If omitted, the timeout time is 120 seconds (2 minutes). Minimum value is 20 seconds.

/TLS

Negotiate with the server to perform TLS authentication as per RFC 4217. The certificate delivered by the server is checked and self signed certificates may be rejected if desired. After performing the negotiation user authentication takes place over an encrypted connection.

Note! Data transfers will not be encrypted until a PROTECTION PRIVATE command has been issued.

/VMS (default)
/**NOVMS**

/VMS negotiates for VMS file structure. /NOVMS does not. If omitted, SET VMS or SET NOVMS determines the outcome (see the SET VMS command for details).

Note! The OPEN /VMS and OPEN /NOVMS settings override SET VMS and SET NOVMS.

Examples

1 The following opens a connection to SYS1. If successful, you have to enter a username and password.

```
FTP>OPEN SYS1
```

2 The following DCL level command opens a connection to SYS1. The line includes the username and password so that you can use the command procedure interactively or in batch processing.

```
$ FTP OPEN SYS1 "smith" "opensesame"
```

3 The following DCL level command opens a connection to SYS1, but uses a Token Authentication PASSCODE derived from the SecurID card, instead of the password:

```
$ FTP OPEN SYS1 "smith" "1234987654"
```

PROTECTION

Set the protection for the data port after doing TLS authentication. RFC 2228 defines CLEAR, PRIVATE, SAFE and CONFIDENTIAL, but RFC 4217 specifies that only CLEAR and PRIVATE can be used with TLS. The PROTECTION command does an FTP PBSZ (protection buffer size) command followed by an FTP PROT command. The PROTECTION should be specified before returning the command channel to clear text mode as the RFCs specify.

Format

PROTECTION level

Parameters

CLEAR

Data transfers take place in the clear, as they would with a traditional FTP session. This is the default if no protection has been specified.

PRIVATE

Data transfers are encrypted such that they cannot be read by an intermediate system and are integrity protected.

Example

FTP>**PROTECTION CLEAR**

FTP>**PROTECTION PRIVATE**

PUT

Copies files to a remote host.

PUT supports full wildcard filespecs except wildcards enclosed in a quoted string. Use the /MULTIPLE qualifier for a wildcarded local-file filespec. PUT also supports use of asterisk (*) wildcards after a semicolon (;) in remote file specifications. This creates the same version in the destination file as in the source file (instead of creating a new version). If the server is not OpenVMS, the version number is part of the filename. TCPware does not issue a warning if the server host already has a higher numbered version.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Select file (or files) in **`Local Files`** ⇒ **`Copy-->`**. Give file new name, if desired, in **`New Remote File/Dir Name`**

Format

PUT *local-file[,local-file,...] [remote-filename]*

Synonyms and Equivalents

COPY *local-file* /LOCAL *remote-filename*
MPUT *wildcarded-local-files [remote-filename]* = PUT *local-file*/MULTIPLE
SEND *local-file[,local-file,...] [remote-filename]*

Parameters

local-file

Input filespec on the local host. Must conform to OpenVMS filenaming rules. Use a comma between multiple filespecs.

The filespec format is:

node"username password"::path

node	hostname or DECnet node name (with OpenVMS Alpha V6.1 and later, and all OpenVMS I64 systems, the host name can be a domain name or IP address)
username	valid account on the host
password	password (PASSCODE if using Token Authentication) for the account
path	location and name of the file

You can omit the node *`"username password"`*`::` part of the specification unless it is for a DECnet file. If omitted, Client-FTP uses the current default directory.

You can use the *`node`*`::`*`path`* syntax (omitting the username and password) if you want access to anonymous FTP resources, in which case the /ANONYMOUS qualifier is implied.

`wildcarded-local-files`

Input filespec on the local host in wildcard format. Wildcards include the percent symbol (%) or the question mark symbol (?) to indicate individual characters, and the asterisk symbol (*) to indicate multiple characters. Examples of wildcarded filespecs are `*.TXT` , `W????.*`, and `*.*;*`.

remote-filename

Output filespec on the remote host. Enclose the filespec in quotes if you want to preserve case and did not use the SET NOLOWERCASE command. If the remote-filename is omitted, Client-FTP uses the *local-file* filename and extension, unless they are part of a quoted string. Also, enclose the filespec in quotes if it contains delimiters or symbols the FTP server can interpret in special ways.

For example, the following remote filespec is enclosed in quotes because it includes slashes (/) OpenVMS normally interprets as qualifier delimiters:

```
ALPHA"smithabcd"::"/usr/bin/proj1.txt"
```

The remote filespec must conform to the filenaming conventions of the remote host. In OpenVMS-to-OpenVMS file transfers, the *local-file* and *remote-filename* specification formats are the same. (See the *local-file* parameter).

To obtain the same version number in the destination file as in the source file (instead of creating a newer one), wildcard the destination file version using `;*`. Note that if the server is not an OpenVMS host, the version number is included in the filename. You do not get a warning if the server host already has a higher numbered version. Also, if the server host already has the version specified, the old file with that version is overwritten.

Qualifiers

If you omit one of the file type qualifiers (/ASCII, /BINARY, /FORTRAN, /IMAGE, or /VMS), Client-FTP transfers the file based on either:

- The current default setting; for example, ASCII or IMAGE.
- The extension (type) of the file you want copied (see Table 3-2).

Setting a file type qualifier with the PUT command overrides the default transfer format for this PUT only.

See also the SET DEFAULT command.

/ANONYMOUS
/NOANONYMOUS

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) anonymous user access to remote resources. You can omit /ANONYMOUS if using the file syntax *`node::path`*. (SeeAnonymous Users.)

/APPEND

Appends the *local-file* file to the *remote-filename*. If the *remote-filename* file does not exist, Client-FTP creates it. Some remote hosts do not support this operation. **NOTE:** If the operation fails, try appending in binary mode by using the /BINARY qualifier.

/ASCII

Transfers the file in formatted ASCII format (see Client-FTP File Transfer Formats).

/BINARY

Transfers .BIN, .LDA, .OBJ, and .STB, files in formatted binary format (see Client-FTP File Transfer Formats).

/**BLOCK**

Transfers STREAM, STREAM_CR, STREAM_LF, and UNDEFINED files in block mode (see Client-FTP File Transfer Formats).

/**CONFIRM**
/NOCONFIRM (default)

/CONFIRM issues a confirmation prompt before putting a file. Respond with **`Y`** or **`N.`** If confirming multiple file puts, use with MPUT or PUT/MULTIPLE with a wildcard value. Position the qualifier immediately after the PUT verb to relate to all files, or after the particular filename to relate to that file only.

/**CONTIGUOUS**=*blocks*

Local output file should have an initial contiguous allocation of the specified number of *blocks*. If the output file is smaller than the specified *blocks,* Client-FTP truncates the number of blocks. If the output file is larger, the additional allocations are non-contiguous. Does not apply to remote output files.

/**CONVERT**
/NOCONVERT (default)

/CONVERT translates the internal file formatting characters of Variable Forms Control (VFC) files. /NOCONVERT does not do the conversion.

/**FDL**

Uses a separate FDL file describing the specified file's OpenVMS RMS record attributes. This qualifier is useful for transferring a VMS node file to a non-VMS node. A subsequent GET /FDL operation can then return the file with the proper record attributes back from the non-VMS node. The default is not to create an accompanying FDL file. The TYPE (or SET TYPE) command determines the type of file. A transfer of:

- ASCII data results in a sequential file with variable records (the default).
- IMAGE data results in a sequential file with fixed length records of 512 bytes.

/**FORTRAN**

Transfers the file in FORTRAN mode (see Client-FTP File Transfer Formats). The first character of each record is a FORTRAN carriage control character. Some hosts do not recognize this transfer format.

/**IGNORE**
/NOIGNORE (default)

/IGNORE ignores errors so that copying can continue with the next file. /NOIGNORE terminates copying if an error occurs.

/**IMAGE***[=size]*

Transfers the file in image mode. Optional size sets the record size of the local output file (see Client-FTP File Transfer Formats). Does not apply to remote output files.

/**LOG**
/NOLOG (default)

/LOG displays file specifications for each file transferred.

/**MULTIPLE**

Transfers multiple files (equivalent to **MPUT**). Use after *local-file* only and include wildcards in *local-file*. Necessary because some remote hosts do not recognize the OpenVMS characters for the asterisk (*), percent (%), or the question mark (?) as wildcards.

/RECORD

Transfers the file using STRU R so as to communicate the record structure during the copy. A positional qualifier. Not all servers support record structure mode. If you specify both /RECORD and /VMS, Client-FTP uses /VMS.

/RESTART

For STREAM mode transfers restart the transfer where it was interrupted. The client verifies that the server supports the RFC 3659 SIZE and REST commands, and ignores the qualifier if it does not.

This does NOT work for VMS mode transfers (STRU VMS), and if the remote system is a VMS system it is recommended that a STRU FILE be done before the transfer command and to include /NOVMS on the command line.

/SET_FACTS

Set selected file facts on the destination file to match the source file after transfer. The facts currently supported are:

MODIFICATION__TIME

/VARIABLE

Transfers an image file (see /IMAGE) in variable length record mode. All /IMAGE records are the same length when stored at the destination. Applies to local output image files only.

/VMS

Transfers the file in VMS file mode (see Client-FTP File Transfer Formats). Allows you to transfer any type of RMS file between OpenVMS systems. /VMS is a positional qualifier. It should immediately follow the filename in question. If you use /VMS, Client-FTP ignores /APPEND, /ASCII, /BINARY, /BLOCK, /FORTRAN, /IMAGE, and /VARIABLE. If you specify both /RECORD and /VMS, Client-FTP uses /VMS. Not all servers support VMS files. If the server does and you do not specify another mode (using a qualifier or the STRUCTURE or SET DEFAULT commands), /VMS is the default.

Examples

1 The following copies the STUFF.TXT file from your local host to the remote host (the receiving system stores the file under the same filename in the default directory):

```
FTP>PUT STUFF.TXT
```

2 The following copies the local STUFF.TXT file to DELTA's anonymous directory. It is equivalent to having used /ANONYMOUS:, sending the *"ANONYMOUS user-email-address"* username and password with the command.

```
FTP>SEND DELTA::STUFF.TXT
```

PWD

Prints the name of the current working directory on the remote host.

Useful for determining the default directory when not specifying a full pathname.

Format

PWD

Equivalent

SHOW DEFAULT

QUOTE

Sends an FTP command to the remote server.

Note! Do not use QUOTE to initiate a file transfer operation.

Format

QUOTE *command*

Equivalents

QUOTE HELP SITE = SITE HELP = HELP /REMOTE SITE = REMOTEHELP SITE

Parameter

command

FTP command string sent to the remote FTP server. FTP commands are not the same as Client-FTP commands. Enclose the command in quotes if it contains special characters or embedded spaces, or is case-sensitive.

Example

The following sends the SYST command to the remote FTP server. If implemented by the remote server, it returns the type of operating system running on the remote server.

```
FTP>QUOTE "SYST"
```

REMOTEHELP

Accesses the remote FTP server's on-line help.

See HELP to bring up Client-FTP's on-line help.

Format

REMOTEHELP *[topic]*

Equivalents

HELP /REMOTE *[topic]*

HELP /REMOTE SITE = REMOTEHELP SITE = SITE HELP = QUOTE HELP SITE

Parameter

topic

Optional topic for which you want help from the remote server. If you do not specify a topic, HELP provides you with a list of topics and prompts you to choose one.

RENAME

Renames a file on the remote host.

Graphical User Interface Equivalent

`TCPware FTP-OpenVMS File Transfers` ⇒ Select file in **`Remote Files`** ⇒ Give file new name in **`New Remote File/Dir Name`** ⇒ **`Rename-->`**

Format

RENAME ***old-name new-name***

Parameters

old-name

File on the remote host to rename. The remote filespec must conform to the filenaming conventions of the remote host. Enclose the filespec in quotes if it contains delimiters or symbols the FTP server can interpret in special ways. If a remote OpenVMS file, the specification is:

`node"username password"::path`

node	hostname or DECnet node name (with OpenVMS Alpha V6.1 and later, and all OpenVMS I64 systems, the host name can be a domain name or IP address)
username	valid account on the host
password	password (PASSCODE if using Token Authentication) for the account
path	location and name of the file

You can omit the *`node"username password"`*`::` part of the specification unless it is for a DECnet file. If omitted, Client-FTP uses the current default directory. You can use the *`node`*`::`*`path`*syntax (omitting the username and password) if you want to rename anonymous FTP resources, in which case the /ANONYMOUS qualifier is implied.

new-name

Valid filespec to substitute for old-name. Enclose in quotes if it contains special characters, imbedded spaces, or is case sensitive.

Qualifier

/ANONYMOUS
/NOANONYMOUS

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) renaming files in anonymous user directories. You can omit /ANONYMOUS if using the node file syntax (*`node`*`::`*`path`*). (See Anonymous Users.)

Examples

1 The following renames the `testb` file to `test2/test`:

FTP>**RENAME testb "test2/test"**

2 The following renames the OLD.TXT file on DELTA to NEW.TXT. It is equivalent to using the /ANONYMOUS qualifier: sends the *`"ANONYMOUS user-email-address"`* username and password with the command.

```
FTP>RENAME DELTA::OLD.TXT NEW.TXT
```

SET [NO]BELL

Enables the terminal bell after completing a file transfer.

SET NOBELL is the default.

Format

SET BELL

SET NOBELL

Synonym

BELL- toggles between SET BELL and SET NOBELL

SET DEBUG /CLASS

Enables or disables displaying debugging information depending on the class keyword(s) used. The /CLASS qualifier is required.

Note! SET DEBUG /CLASS=REPLIES (or VERBOSE toggled to ON) is the default. In this way, you can see informational messages when logging in to the server or changing remote directories (if informational messaging is enabled on the server).

Format

SET DEBUG /CLASS=*(keyword,...)*

Synonyms

DEBUG - toggles SET DEBUG /CLASS=COMMANDS

VERBOSE - toggles SET DEBUG /CLASS=REPLIES (default is ON)

Qualifier

/CLASS=*(keyword,...)*

Classes of debugging information to enable or disable. Use one or more of the keywords listed in Table 3-8. The initial default is PERFORMANCE and REPLIES. Use NONE as the first entry to clear the classes before resetting them (see Example 1).

Table 3-8 Class Keywords

Keyword	**Purpose**
COMMANDS	Enables displaying FTP commands sent to the server.
PERFORMANCE	Enables displaying performance information (when using COPY/LOG, GET/LOG, or PUT/LOG).
REPLIES	Enables displaying FTP replies received from the server; equivalent to toggling the VERBOSE command ON (the default).
ALL	Enables displaying all classes.
NONE	Disables displaying all classes.

Examples

1 The following resets the debugging classes. It first disables all classes (NONE), and then enables the COMMANDS and REPLIES (VERBOSE) classes.

```
FTP>SET DEBUG/CLASS=(NONE,COMMANDS,REPLIES)
```

2 The following toggles the REPLIES (VERBOSE) class. If on, it shows informational messages (if enabled on the server) when logging in or moving around directories on the server. The ON or OFF setting is immediately displayed after the command.

```
FTP>VERBOSE
```

SET DEFAULT

- Changes the default local or remote directory
- Sets the default qualifiers used with the COPY, GET, PUT, and DELETE commands

Note! Specify the parameter or the qualifiers separately. Do not specify them together.

Format

SET DEFAULT *[directory]*

Synonyms and Equivalents

CD *[directory]*= SET DEFAULT /REMOTE (CD allows you to use UNIX-style *directory* names)

LCD *[directory]*= SET DEFAULT /LOCAL

IMAGE = SET DEFAULT /IMAGE

TYPE BINARY = SET DEFAULT /BINARY

Parameter

directory

Default directory to set on the local or remote host, depending on whether the /LOCAL or /REMOTE qualifier follows, or the remote directory specification if no qualifier follows. The directory format is:

```
[node"username password"::]directory
```

To open a connection first, use the `node"username password"::` part of the format. This syntax is optional. The *directory* part of the format is any valid directory specification. Enclose it in quotes if it contains special characters or embedded spaces, or is case-sensitive. (You can also use the **`[directory]`** format, as in [-], if the remote host is an OpenVMS system.) If *directory* is omitted:

- With SET DEFAULT or SET DEFAULT /REMOTE, Client-FTP sets the default directory to the parent of the current directory on the remote host.
- With SET DEFAULT /LOCAL, Client-FTP sets the local default directory to your login directory defined by the SYS$LOGIN logical.

Use the `node::directory` syntax to access an anonymous FTP user directory, in which case you can omit the /ANONYMOUS qualifier.

Qualifiers

/LOCAL

Changes the local default directory to directory. LCD is the same as SET DEFAULT /LOCAL.

/REMOTE (default)

Changes the remote default directory to directory. CD is the same as SET DEFAULT /REMOTE.

/ANONYMOUS
/NOANONYMOUS

Enables (/ANONYMOUS) or denies (/NOANONYMOUS) the setting of defaults for anonymous user directories. You can omit /ANONYMOUS if you use the syntax `node::directory`. (See Anonymous Users.)

/[NO]**APPEND**
/[NO]**CONFIRM**
/[NO]**IGNORE**
/[NO]**LOG**
/[NO]**RECORD**
/[NO]**VARIABLE**
/[NO]**VMS**

These qualifiers set various transfer defaults. Do not use with /LOCAL or /REMOTE. See the COPY, GET, PUT, or DELETE command for qualifier descriptions.

/**ASCII**
/**BINARY**
/**BLOCK**
/**FORTRAN**
/**IMAGE***[=n]*

These qualifiers set transfer mode defaults (see Client-FTP File Transfer Formats). Use only one. Do not use with /LOCAL or /REMOTE. See the COPY, GET, or PUT command for qualifier descriptions.

/DEFAULT

Determines the default transfer mode from the local file's file extension. Do not use with /LOCAL or /REMOTE.

Examples

1 The following equivalent commands set the local default directory to [SMITH.DOC]. The default device does not change.

```
FTP>SET DEFAULT /LOCAL [SMITH.DOC]
FTP>LCD [SMITH.DOC]
```

2 The following equivalent commands sets the remote default directory to `/usr/src/`:

```
FTP>SET DEFAULT /REMOTE "/usr/src/"
FTP>CD "/usr/src/"
```

3 The following sets the default transfer mode to /IMAGE for subsequent copy commands, and sets the default to /LOG and /NOCONFIRM:

```
FTP>SET DEFAULT /IMAGE /LOG /NOCONFIRM
```

4 The following sets the remote directory to the anonymous directory on DELTA.

```
FTP> SET DEFAULT DELTA::[]
```

It is equivalent to:

```
FTP>SET DEFAULT DELTA"ANONYMOUS user-email-address"::[]
```

5 The following sets the remote directory to SYS$SYSDEVICE:[USER.SMITH]:

```
FTP>CD "/sys$sysdevice/user/smith"
```

SET DEFLATE

- Changes the options for MODE DEFLATE transfers. The only deflate engine present is ZLIB.

- DEFLATE transfers can be enabled with the SET MODE DEFLATE command. DEFLATE transfers cannot be used when TLS is being used.

Format

SET DEFLATE

Qualifiers

/LEVEL={-1-9}

Changes the level of data compression that the engine uses when a file is transferred. The default level is -1 a compromise between speed and compression, 0 is no compression, 1 is best speed and 9 is best compression.

SET *[*NO*]*ALLOWSELFSIGNED

Allows or disallows self signed certificates for RFC 4217 TLS negotiation.

The default is to allow self signed certificates.

Format

SET ALLOWSELFSIGNED

SET NOALLOWSELFSIGNED

SET *[*NO*]*HASH

Enables hash marks.

With SET HASH, Client-FTP displays a hash mark (#) every 1024 bytes sent or received during a file transfer. SET NOHASH is the default.

Hash marks appear in files only. No hash marks appear if the file transfer is output to the terminal screen.

Note! With SET HASH, FTP reads only 1024 bytes at a time from the network layer. While this means that FTP gives more accurate reports on the progress of a transfer, it increases overhead. Use hash marks primarily with transfers over slower-speed links (such as SLIP lines).

Format

SET HASH
SET NOHASH

Synonym

HASH- toggles between SET HASH and SET NOHASH

SET *[*NO*]*LOWERCASE

Enables the conversion of unquoted filenames to lowercase before Client-FTP sends the files to the remote host. SET LOWERCASE is the default.

With SET NOLOWERCASE, Client-FTP does not convert unquoted filenames to lowercase.

Note! Client-FTP always preserves the case of filenames that appear within quotation marks.

Format

SET LOWERCASE
SET NOLOWERCASE

SET MODE

Set the transfer mode to STREAM (default), BLOCK, COMPRESSED or DEFLATE

Format

SET MODE {STREAM | BLOCK | COMPRESSED | DEFLATE}

Parameters

STREAM

The data are transmitted as a stream of bytes. This is the default.

BLOCK

The file is transmitted as a series of data blocks preceded by one or more header bytes.

COMPRESSED

Data that contains repeated sequences may be compressed to obtain better bandwidth.

DEFLATE

The data is compressed with the ZLIB compression engine. DEFLATE cannot be used with sessions that use TLS authentication. The TLS code provides data compression when the data stream is protected.

Example

FTP>**SET MODE STREAM**

FTP>**SET MODE DEFLATE**

SET *[*NO*]*PASSIVE

Sets passive mode. Passive mode performs an active open on the data connection, which can avoid problems with firewall systems.

/SET NOPASSIVE (the default) disables passive mode.

Note! You can also define the TCPware FTP_PASV logical as follows:

```
$ DEFINE/PROCESS TCPWARE_FTP_PASV "TRUE"
```

Your system manager can also define the logical system-wide as follows:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_PASV "TRUE"
```

Graphical User Interface Equivalent

TCPware FTP-OpenVMS File Transfers ⇒ Options ⇒ TCPware FTP-OpenVMS Settings ⇒ PASV Mode⇒OK

Format

SET PASSIVE
SET NOPASSIVE

Synonym

PASSIVE - toggles between SET PASSIVE and SET NOPASSIVE

SET *[*NO*]*VMS

Controls whether the Client-FTP negotiates for VMS file structure with the FTP server when opening a connection. The default is SET VMS, where the client negotiates with the server to use File Descriptor Language (FDL) information.

Client-FTP first queries if the server supports VMS file transfer mode. If not, it queries for VMS Plus file transfer mode, such as with HP's TCP/IP Services for OpenVMS (UCX) server.

In connecting to a TCPware or other OpenVMS server, the VMS file structure transfer mode is used.

See Client-FTP File Transfer Formats for more information.

Note! OPEN /VMS or OPEN /NOVMS overrides SET VMS and SET NOVMS.

Format

SET VMS (default)
SET NOVMS

SHOW STATUS

Displays the following information about your present FTP session:

- Remote hostname and internet address if you are connected to a remote host
- Username on the remote host if you are connected and logged in
- Local default directory
- Remote default directory if you are logged in to a remote host and that host supports the FTP PWD command
- Record size to be used with the /IMAGE qualifier
- Defaults that are defined by the SET DEFAULT command for the COPY, GET, PUT, and DELETE commands

Format

SHOW STATUS

Synonym

STATUS

Example

The following shows the status for the current connection:

```
FTP>SHOW STATUS

Connected to ALPHA (192.168.1.1)
Logged in as user "SMITH"

The local default is SYS$COMMON:[SYS$LDR]
The remote working directory is /usr/users

Default qualifiers are /VMS
```

SITE

Issues a site-specific command to the remote server.

Format

SITE *command*

Equivalents

SITE HELP = HELP /REMOTE SITE = REMOTEHELP SITE = QUOTE HELP SITE

Parameter

command

Site-specific command string to send to the remote host. Enclose the command in quotes if it contains special characters or embedded spaces, or is case sensitive. Site-specific commands can vary depending on the remote FTP server; some servers do not support any.

This command is often useful in obtaining information about the site-specific commands, if any, the remote FTP server supports.

Example

The following sends a site-specific command (SITE SPAWN PRINT MYFILE.TXT) to the remote server. With the FTP-OpenVMS server, requests printing of the MYFILE.TXT file.

```
FTP>SITE "SPAWN PRINT MYFILE.TXT"
```

SPAWN

Executes DCL commands without exiting FTP.

Note! Spawning is not allowed for CAPTIVE accounts.

Format

SPAWN *[command-line]*

Parameter

command-line

DCL command line you want executed. If omitted, spawns an interactive subprocess. To return from an interactive subprocess, enter **LOGOUT**.

Synonym

Z *[command-line]*

Examples

1 The following displays the time on your local host without leaving Client-FT:

```
FTP>SPAWN SHOW TIME
  3-NOV-2014 14:02:48
```

2 The following initiates DCL command mode, displays the local time, logs out, and returns to Client-FT:

```
FTP>SPAWN
$ SHOW TIME
  3-NOV-2014 14:02:51
$ LOGOUT
  Process SMITH_1 logged out at 3-NOV-2014 14:02:54.34
FTP>
```

STRUCTURE

Changes the default file structure.

Client-FTP uses FILE structured files as the default. Use the /[NO]RECORD qualifier for the COPY, GET, or PUT commands to override this default for individual transactions.

Format

STRUCTURE *keyword*

Parameter

keyword

Table 3-9 lists valid values for *keyword*.

Table 3-9 STRUCTURE Command Keyword Values

Value	Purpose
FILE	Sets FILE as the default file structure. FILE structured files consists of sequential bytes. Equivalent to SET DEFAULT/NORECORD. This is the default.
RECORD	Sets RECORD as the default file structure. RECORD structured files consists of a collection of records. Equivalent to SET DEFAULT/RECORD.
VMS	Sets VMS as the default file structure. VMS file structure allows you to transfer all types of RMS files between OpenVMS systems using File Descriptor Language (FDL) information. May OpenVMS systems that implement FTP support this structure. Equivalent to SET DEFAULT/VMS.

Note! Some FTP servers do not support the RECORD or VMS structures.

Example

The following changes the default file structure to FILE:

```
FTP>STRUCTURE FILE
```

TYPE

Changes the default file transfer format for all future file operations in this session.

The following rules apply to the TYPE command:

- The default file transfer format remains set until you redefine it. It does not change when opening or closing a connection.
- The default format changes only if the remote host accepts the type change.
- If there is no default file format defined, Client-FTP tries to determine the file format based on the local file's file extension.

Use the COPY, GET, or PUT command qualifiers to override this default for individual transactions.

Format

TYPE *keyword*

Equivalents

SET DEFAULT type *qualifier*

ASCII= TYPE ASCII

BINARY= TYPE IMAGE

IMAGE= TYPE IMAGE

Parameter

keyword

Table 3-10 lists valid values for keyword. See Client-FTP File Transfer Formats for a full description of the file transfer types.

Table 3-10   TYPE Command Keyword Values

Keyword	**Purpose**
ASCII	Sets formatted ASCII format (see Client-FTP File Transfer Formats). Equivalents: • SET DEFAULT/ASCII • ASCII
BINARY	Sets formatted binary format (see Client-FTP File Transfer Formats). SET DEFAULT/BINARY is equivalent.
IMAGE	Sets image format (see Client-FTP File Transfer Formats). Equivalents: • SET DEFAULT/IMAGE • BINARY • IMAGE

FORTRAN	Sets ASCII format and specifies that the first character of each record is a FORTRAN carriage control character (see Client-FTP File Transfer Formats). SET DEFAULT/FORTRAN is equivalent.
BLOCK	Sets block format (see Client-FTP File Transfer Formats). SET DEFAULT/BLOCK is equivalent.
VARIABLE	Specifies that FTP writes an image format file as a variable-length record format file. Although FTP writes the records as variable-length, all records are the same length. SET DEFAULT/IMAGE/VARIABLE is equivalent.
DEFAULT	Removes the previous default file format. SET DEFAULT/DEFAULT is equivalent. This is the default setting for an undefined format.

Examples

1 The following changes the default file format to formatted ASCII:

```
FTP> TYPE ASCII
```

2 The following removes the previous default file format. For future transactions, Client-FTP tries to determine the file format based on the local file's extension.

```
FTP>TYPE DEFAULT
```

USER

Sets the username at the remote host.

USER requires an open connection.

Format

USER *[username [password [account]]]*

If you:

- Supply the username, password, and account (if required) with the command, you are not prompted for them separately.
- Omit the parameters from the command line, you are prompted for them.
- Use USER in an interactive command file and do not want to be prompted for a user name, enter the username in the file on the line after the USER command. (You cannot include password or account information in the interactive command file.)
- Use the command non-interactively (for example, a batch job), and do not want to be prompted for a username, password, or account, then include the parameters on subsequent lines, after the USER command, in the command file.
- Want to be prompted for a password, do not use the command file with a batch job nor specify the password in a command file.

The display does not echo the password or account information.

Synonym

LOGIN

Parameters

username

Username on the remote host. Enclose the username in quotes if case is important or if it contains special characters. Prompted if omitted.

password

Password on the remote host. Enclose the password in quotes if case is important or if it contains special characters. Prompted if omitted and required. Not echoed.

If you are designated by the system administrator as having password authentication through Token Authentication, you need to enter the PASSCODE in place of the password. Depending on which type of SecurID card you were assigned:

- Enter a combination of your memorized personal identification number (PIN) and the tokencode that appears on the card (with no separating space) as the password, or
- Enter your memorized PIN on the PINPAD] card and the resulting tokencode that appears on the card as the password.

See the Chapter 15, *Token Authentication: Protecting Logins*, for details on obtaining PASSCODEs.

account

Account on the remote host. Enclose the account in quotes if case is important or if it contains special characters. Prompted if omitted and required. Not echoed.

Example

The following sets the username on the remote host to SMITH, and specifies a password and an account:

```
FTP>USER "SMITH" "PASSWORD" "SMITH"
```

Chapter 4 Kerberos User Commands

Introduction

This chapter describes the user functions needed to get a ticket-granting ticket for Kerberos applications and maintaining the ticket file.

Ticket File Location Logical

The default ticket file for the user is SYS$LOGIN:KERBV4.TICKET. If you define the TCPWARE_KERBV4_TKFILE logical, you can have the ticket file located somewhere else.

Here is an example of how you can define this logical to locate the ticket file in a specific directory:

```
$ DEFINE/PROCESS TCPWARE_KERBV4_TKFILE SYS$DISK:[MYDIR]TICKET.TXT
```

Kerberos User Commands

The user interface with Kerberos comprises the following commands:

GET TGT	Gets the ticket-granting ticket (TGT) to authenticate yourself to Kerberos
REMOVE TICKETS	Removes the TGT and any service tickets you might have
SET KERBEROS_PASSWORD username	Changes your Kerberos password
SHOW TICKETS	Lists all tickets (TGT and service tickets) in the ticket file

You can access these commands using the Network Control Utility (NETCU) by entering either:

-
  ```
  $ NETCU
  NETCU>command
  ```

-
  ```
  $ NETCU command
  ```

Uppercase command parameters are converted to lowercase unless you enclose them in quotes.

Command Reference

A description of each Kerberos user command follows.

GET TGT

For Kerberos users. Gets the ticket-granting ticket (TGT) that allows you to get application service tickets. This process authenticates you to the Kerberos Server, which is considered to be a trusted, secure machine. TGTs are required to obtain an application service ticket from the Kerberos Server.

The name of the ticket file is determined by the TCPWARE_KERBV4_TKFILE logical, usually set to SYS$LOGIN:KERBV4.TICKET.

You must enter your Kerberos password with this command.

Your OpenVMS login name is used for the Kerberos username unless the /USERNAME qualifier specifies otherwise.

GET TGT is equivalent to the UNIX command `kinit`.

Format

GET TGT

Password: ***password***

Parameter

password

User's Kerberos password that authenticates the user to the Kerberos Server. Converted to lowercase unless you enclose it in double quotes.

Qualifiers

/INSTANCE=*instance*

Usually omitted for a general Kerberos user; **`admin`** for an administrative user. (See your Kerberos administrator to determine your Kerberos instance name.) Converted to lowercase unless you enclose it in double quotes.

/LIFETIME=*minutes*

Lifetime of the TGT in minutes ranging from 5 to 1275 minutes. The default lifetime is 480 minutes (8 hours).

/REALM=*realm*

Optional Kerberos realm to use instead of the one determined by the value of the logical TCPWARE_KERBV4_REALM. Converted to lowercase unless you enclose it in double quotes.

/USERNAME=*login-name*

Alternate login name. Converted to lowercase unless you enclose it in double quotes.

Example

```
NETCU GET TGT
Password:
```

Gets a ticket-granting ticket for the logged-in user. If the user logged in as SYSTEM, SYSTEM is used as the Kerberos username—if the user logged in as FRED, FRED is used as the Kerberos username.

REMOVE TICKETS

For Kerberos users. Removes your ticket-granting ticket and application service tickets, if any.

See the SHOW TICKETS command to view the user's ticket-granting ticket and any application service tickets contained in the user's ticket file.

The name of the ticket file is determined by the value of the TCPWARE_KERBV4_TKFILE logical, usually set to SYS$LOGIN:KERBV4.TICKET.

REMOVE TICKETS is equivalent to the UNIX command `kdestroy`.

Format

REMOVE TICKETS

Qualifiers

/BELL
/NOBELL (default)

Specifies whether the terminal bell should sound when an error occurs when trying to remove tickets. The default is /NOBELL.

/STATUS (default)
/NOSTATUS

Specifies whether to display a message when removing tickets. The default is /STATUS.

Example

```
NETCU REMOVE TICKETS
```

Removes the ticket-granting ticket and application service tickets, if any.

Troubleshooting

```
%TCPWARE_NETCU-W-NTKTTODES, no tickets to destroy
```

Meaning: The ticket file does not exist.

Action: Use the GET TGT command to create a ticket file entry.

```
%TCPWARE_NETCU-I-TKTDESTR, tickets destroyed
```

Meaning: The ticket was successfully removed.

```
%TCPWARE_NETCU-E-TKTNODES, tickets NOT destroyed
```

Meaning: Some error occurred while trying to delete the ticket file. Possible reasons are that the ticket file does not grant delete access, or you are not its actual owner.

SET KERBEROS_PASSWORD

For Kerberos users. Changes your Kerberos password.

Note! If you change your Kerberos password, your ticket-granting ticket (TGT) is deleted from your ticket file. You need to create a new TGT using the GET TGT command.

SET KERBEROS_PASSWORD is equivalent to the UNIX command `kpasswd`.

Format

SET KERBEROS_PASSWORD *[username [instance]]*

Old password for username: ***old-password***

New password for username: ***new-password***

Verifying, please re-enter: ***new-password***

Parameters

username

Kerberos username for which to change the Kerberos password. If omitted, the OpenVMS username under which the user logged in is used. Converted to lowercase unless you enclose it in double quotes.

instance

Usually omitted for a general Kerberos user but can be the name of the machine from which you can obtain ticket-granting tickets and service tickets. Specify **`admin`** for an administrative user. (See your Kerberos administrator to determine your Kerberos instance.) Converted to lowercase unless you enclose it in double quotes.

old-password
new-password

Old and new user passwords. Converted to lowercase unless you enclose them in double quotes.

Example

```
NETCU SET KERBEROS_PASSWORD PERSEPHONE
Old password for 'persephone':
New password for 'persephone':
Verifying, please re-enter:
```

Changes the Kerberos password for user `persephone`.

SHOW TICKETS

For Kerberos users. Displays your ticket-granting ticket (TGT) and any existing application service tickets.

The name of the ticket file is determined by the value of the TCPWARE_KERBV4_TKFILE logical, usually set to SYS$LOGIN:KERBV4.TICKET.

SHOW TICKETS is equivalent to the UNIX command `klist`.

See the GET TGT command for more information on getting ticket-granting tickets.

Format

SHOW TICKETS

Qualifiers

/BRIEF
/NOBRIEF (default)

/BRIEF lists only the acquired tickets and not the ticket files, principal names, issuance dates, or expiration dates.

/SRVTAB

Shows the contents of the `TCPWARE:SRVTAB`. file as a list of available Kerberos services. (See CREATE SRVTAB for more information on the `TCPWARE:SRVTAB`. file.)

/TGT_TEST
/NOTGT_TEST (default)

Checks whether the tickets are still valid and returns a success or failure exit status.

Examples

1

```
NETCU SHOW TICKETS
Ticket file:    SYS$LOGIN:KERBV4.TICKET
Principal:      fred@daisy.com

Issued          Expires         Principal
----------------------------------------------
Jun 1 10:11:12  Jun 1 18:11:12  krbtgt.daisy.com@daisy.com
```

Displays the name of the ticket file; ticket owner's principal name, issue and expiration dates; and service principal name of each ticket.

2

```
NETCU SHOW TICKETS /SRVTAB
Server key file:   TCPWARE:SRVTAB.
Service        Instance        Realm        Key Version
------------------------------------------------------------
changepw       bart            daisy.com    1
rcmd           bart            daisy.com    1
```

Lists the available Kerberos services on BART as listed in its `TCPWARE:SRVTAB`. file.

Chapter 5 Network Printing

Introduction

The TCPware for OpenVMS network print services include Line Printer Services (LPS) and Terminal Server Print Services. These network printing services support most printing devices, including line printers, laser printers, and plotters.

TCPware provides Internet Printing Protocol support. refer to the *TCPware for OpenVMS Management Guide* for more information about IPP.

LPS lets users print files on printers attached to remote hosts. Users can also print files that are on a remote host to printers attached to the local host.

Terminal Server Print Services lets users print files on printers attached to terminal servers on a TCP/IP network.

TCPware bases the network printing services on:

- UNIX style LPR/LPD protocols—Line Printer Services (LPS) implement these protocols. LPS supports the UNIX style LPR, LPRM, and LPQ commands, and the OpenVMS style PRINT command. You can configure a host as an LPS client and an LPS server (LPD).

  The LPS OpenVMS print queue created during configuration can be a queue that:

  – Performs local OpenVMS print formatting and prints output on the printer associated with the remote host running LPD.

  – Sends local print requests to the remote print queue running LPD. The remote print queue performs the print formatting.

- OpenVMS print protocol—Terminal Server Print Services implements this protocol and supports the OpenVMS style PRINT command.
- Before you use the TCPware network printing services, get a list of available print queue names from your system manager and be sure that:
- TCPware print services software has been configured and started on your system.
- Any other required OpenVMS print queues have been initialized and started.

Network Print Services

Once the print queue has been initialized and started, you can send print requests to a printer attached to a remote host, or to a printer connected to a terminal server on the TCP/IP network. You can also print files that are on a remote host to printers attached to the local host.

The LPS client and Terminal Server Print Services support the following commands:

LPQ	Displays the remote print job status	**LPRM**	Removes a job from a remote print queue
LPR	Sends a job to the default remote printer designated during configuration	**PRINT**	Places a job in the designated print queue; then sends the job to the printer associated with that queue.

Figure 5-1 shows using the UNIX style LPR command and the OpenVMS style PRINT command when you use LPS. It also shows sending a file to a print queue associated with a terminal server on a TCP/IP network.

To send files to a printer using the networking print services:

1 Enter the **`LPR`** command to send a file to print when either the local or remote host is a UNIX system. For example:

```
LPR filename
```

Prints the file specified by *filename* on the default remote printer. For example:

`LPR MEMO.TXT`	Prints the file `MEMO.TXT` on the default remote printer.
`LPR -PMYUNIX MEMO.TXT`	Sends the file MEMO.TXT to the remote printer specified by the logical `MYUNIX`.
`LPR -PRPRINTER1@ALPHA MEMO.TXT`	Sends the file MEMO.TXT to the remote printer `RPRINTER1` connected to host `ALPHA`. See the LPR, LPQ, LPRM, and PRINT commands in the command reference.

2 Enter the PRINT command to send a file to a print queue for printing when one of the following is true (see Figure 5-1):

a The local host is a TCP/IP OpenVMS host and the remote host runs the LPD server.

b The local and remote hosts are TCP/IP OpenVMS hosts.

c The local host is a TCP/IP OpenVMS host and the printer connects to a terminal server on a TCP/IP network.

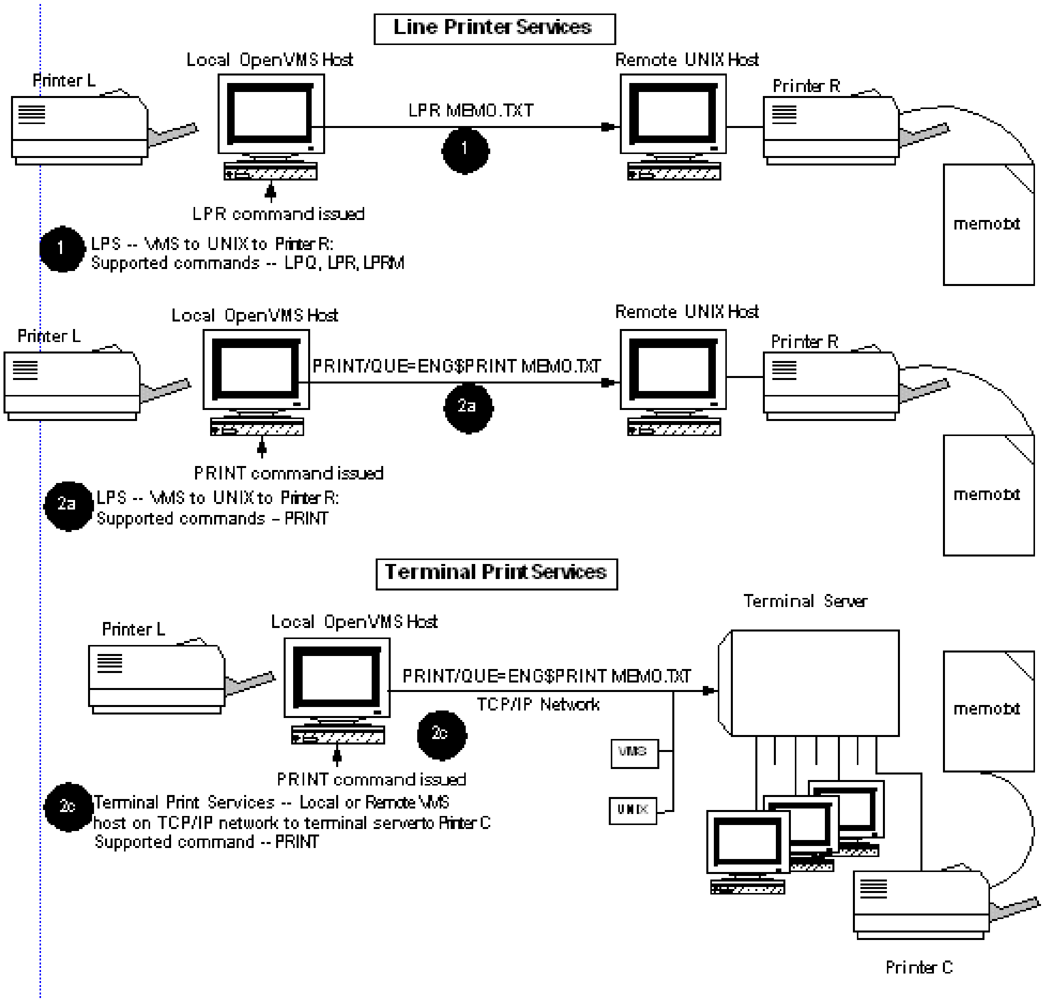
In the print request **`PRINT/QUEUE=`*`qname filename`***, the *qname* parameter is the name of the print queue, and the *filename* parameter specifies the data file or files you want printed.

For example, the print request **`PRINT/QUEUE=ENG$PRINT MEMO.TXT`** sends the file `MEMO.TXT` to the remote printer `queue ENG$PRINT` for printing on the printer associated with that print queue.

The standard OpenVMS qualifiers for the PRINT/QUEUE command are available.

See HP's OpenVMS DCL Dictionary for details on the PRINT command.

Figure 5-1 Using UNIX Style and OpenVMS Style Printing Commands



PRINT Qualifiers

LPS supports the OpenVMS PRINT/FORM qualifier on local LPS OpenVMS print queues. LPS OpenVMS print queues configured with the VMS formatting option support the /FORM qualifier.

LPS also supports the /PARAMETERS qualifier on remote hosts associated with the local LPS OpenVMS print queue. OpenVMS print queues configured with the LPD formatting option support the /PARAMETERS qualifier.

LPS also supports the /LIBRARY and other qualifiers associated with the OpenVMS INITIALIZE/QUEUE command. You can specify these qualifiers during CNFNET configuration.

Figure 5-2 shows the effects of using the /FORM and /PARAMETERS qualifiers on an LPS OpenVMS queue configured for:

- **`VMS`** formatting option set up during configuration—use the /FORM qualifier
- **`LPD`** formatting option set up during configuration—use the /PARAMETER qualifier

If you intend to use the /FORM or /PARAMETER qualifier:

- The format of the PRINT command with the /FORM option is:

`PRINT/QUEUE=`*`qname filename`*` /FORM=`*`form`*

  qname is the OpenVMS queue name and *form* is the form name or number.

  Use the SHOW QUEUE/FORM command to display the list of the available forms for use with LPS.

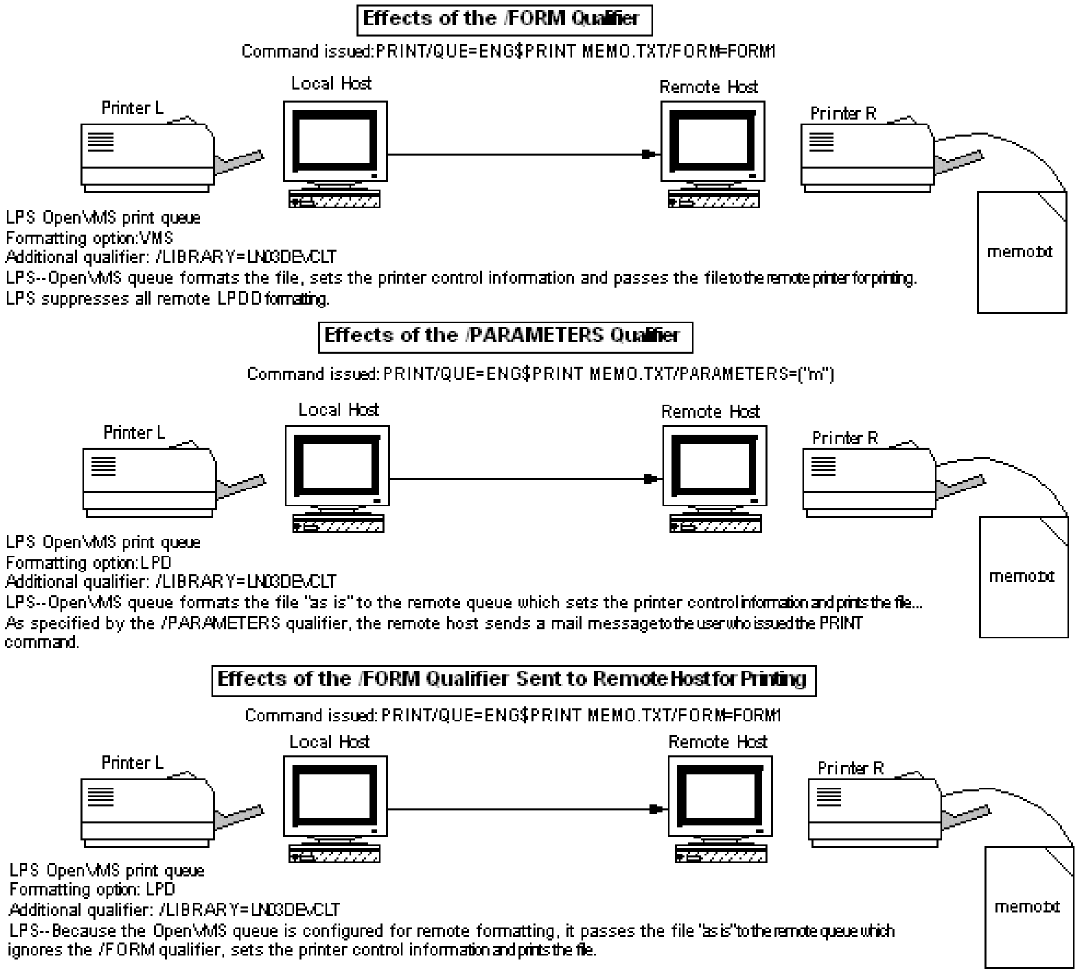
- The format of the PRINT command with the /PARAMETERS option is:

`PRINT/QUEUE=`*`qname filename`*` /PARAMETERS=(`*`parameters`*`)`

  qname is the OpenVMS queue name and *parameters* is any of a number of supported parameters and their values, separated by commas, such as /PARAMETERS=(SIDES=2,NUMBER_UP=2), which indicates double-sided printing with two print "frames" (the "number up") per page.

- Ask your system manager for a list of LPS OpenVMS print queues that support these qualifiers.
- Ask your system manager for a list of available forms for LPS.

Figure 5-2 /FORM and /PARAMETERS Qualifiers with LPS OpenVMS Print



Effects of the /FORM Qualifier

Command issued: PRINT/QUE=ENG$PRINT MEMO.TXT/FORM=FORM1

LPS OpenVMS print queue
Formatting option:VMS
Additional qualifier: /LIBRARY=LN03DEVCLT
LPS--OpenVMS queue formats the file, sets the printer control information and passes the file to the remote printer for printing.
LPS suppresses all remote LPDD formatting.

Effects of the /PARAMETERS Qualifier

Command issued: PRINT/QUE=ENG$PRINT MEMO.TXT/PARAMETERS=("m")

LPS OpenVMS print queue
Formatting option:LPD
Additional qualifier: /LIBRARY=LN03DEVCLT
LPS--OpenVMS queue formats the file "as is" to the remote queue which sets the printer control information and prints the file...
As specified by the /PARAMETERS qualifier, the remote host sends a mail message to the user who issued the PRINT command.

Effects of the /FORM Qualifier Sent to Remote Host for Printing

Command issued: PRINT/QUE=ENG$PRINT MEMO.TXT/FORM=FORM1

LPS OpenVMS print queue
Formatting option: LPD
Additional qualifier: /LIBRARY=LN03DEVCLT
LPS--Because the OpenVMS queue is configured for remote formatting, it passes the file "as is" to the remote queue which ignores the /FORM qualifier, sets the printer control information and prints the file.

LPQ

Displays the status of specific print requests or all requests in a remote print queue.

For each request in a queue, LPQ reports the following:

- User's name
- Current rank of the request in the queue
- Names of the files within the request
- Request number
- Total size of the request in bytes

Print requests appear in the order in which you want them printed. If the filenames are unavailable (because the job consists of text entered directly from the keyboard), LPQ lists them as SYS$INPUT.

You can specify up to 50 files and 50 usernames on one LPQ command line.

LPQ *[-l][-*P*printer] [job-number...] [username...]*

You can enter commands, parameters, and options in upper or lowercase letters. Print services converts all uppercase letters to lowercase unless you enclosed them in quotation marks ("").

Parameters

job-number

Displays queue information for the specified request.

username

Displays queue information for print requests owned by a specific user.

Options

-l

Displays queue information in the long format. If you do not use this option, LPQ displays only as much information about the job as fits on one line.

-P*printer@host*
-P*logical-name*

Specifies a remote print queue. If you do not use this option, LPQ displays information only for the default printer defined by the logical TCPware LPR_PRINTER.

Note! LPQ does not support the UNIX LPQ option `+n`.

Examples

1 This command displays in short form all jobs queued to the printer `sys$print` on host `daisy` and owned by user `smith`:

```
LPQ
lp is ready and printing
Rank     Owner      Job    Files                Total Size
active   smith      45     memo1, memo2, memo   3957 bytes
1st      jones      46     prog.c                897 bytes
2nd      ross       47     letter.txt            432 bytes
```

2 This command displays in long form all jobs queued to the printer `sys$print` on host `daisy` and owned by user `smith`:

```
LPQ -l -Psys$print@daisy smith
lp is ready and printing
smith: active            [job 045daisy.flower.com]
   3 copies of memo.txt             957 bytes

smith:1st                [job 046daisy.flower.com]
   prog.c                           897 bytes

smith:3rd                [job 048daisy.flower.com]
   letter.txt                       432 bytes
```

3 This command displays job 489 in the queue for the default remote printer:

```
LPQ 489
lp is ready and printing
Rank    Owner     Job     Files                Total Size
active  gordon    489     aug.txt, sept.txt    560 bytes
```

LPR

Sends a file to a remote print queue.

If you omit a filespec, the job consists of data you type from the keyboard.

The TCPWARE_LPR_PRINTER logical defines the default remote printer.

TCPware creates the LPR temporary file in SYS$SCRATCH. In this way, if you have a limited disk quota, you can print by redefining the SYS$SCRATCH logical to point to a public scratch disk that has no disk quota limitations.

Format

LPR *[option...] [filespec ...]*

Parameter

filespec

Name of the file(s) you want queued. Use the asterisk (`*`) or percent sign (`%`) as a wildcard character. Enclose in quotes if you want to preserve case other than all lowercase. For multiple files, leave a space between each *filespec*. The default extension is .LIS.

Options

Note! The following options are listed in the order characters (lowercase and uppercase), *numbers*, and *symbols*. They are all prefixed by a hyphen and some take arguments. The lowercase and uppercase character options can mean very different things and are listed together for comparison sake. The important distinction is that the uppercase options all take arguments. There are two ways to keep this distinction clear on the command line:

Enter lowercase options in lowercase and uppercase options in uppercase	Here you MUST enclose the uppercase character in quotes; for example, **`-"P"`** ("use the remote printer indicated by the following argument"). Also include a space character between a lowercase (unquoted) option and any *filespec*, or the entry will be interpreted as an option that takes an argument (see the next method).
Enter all options in lowercase	Here you MUST distinguish the options taking arguments by appending the argument *immediately* after the option character (with no intervening space). For example,**`-plp`** means "use remote printer `lp`", while **`-p lp`** (with the space) means "print the `lp`*file*, which contains UNIX `pr` formatting commands."

Note!

-c

File contains data in the UNIX CIF graphics language.

-"C"*job-classification* (or -**c***job-classification*)

Names the job classification you want used on the burst page. If you omit this option, the job classification is the domain name of the local host. (See the previous note for details on syntax.)

-d

File contains output from TeX formatting commands.

-f

Uses a filter that interprets the first character of each line in the file as a standard FORTRAN carriage control character.

-g

File contains standard UNIX plot data as produced by the plot routines.

-h

Suppresses the printing of the burst page.

-i*[number]*

Indents the output the specified number of blank spaces. If you do not enter a *number*, the output indents eight spaces. (Do not leave a space between the `-i` and the *number*.)

-"J"*job* (or **-j***job*)

Prints the job name on the burst page. If you do not use this option, the job name is the name and extension of the first file in the job. (See the Note for details on syntax.)

-l

Uses a transparent filter so that you can send data to the printer unchanged. Note that the data is UNCONVERTED; print services does not convert the files to STREAM-LF format. Use this option with BINARY data or files containing all of the characters, including carriage returns (CRs), when you want them sent to the printer.

-m

Sends a mail message to the user who issued the LPR command upon completion of the job. You can use this option only if your local host implements the Simple Mail Transfer Protocol (SMTP). This also sets the /NOTIFY option for PRINT, so that if you are logged in as the user under which the job was printed, you will be notified that the job completed.

-n

File contains UNIX `ditroff` (device independent `troff`) formatting commands.

-o

File contains PostScript input.

-p

Prints the file with page headers. (Do not append any characters onto the p of the option or it can be interpreted as an argument to the uppercase -P option. See the Note.)

-"P"*printer@host* (or **-p***printer@host*)

-"P"*logical* (or **-p***logical*)

Specifies a remote printer. If you do not use this option, lpr uses the default printer defined by the logical TCPWARE_LPR_PRINTER. (See the Note for details on syntax.)

-r

Deletes the files from your local host after sending them to the remote queue. Use this option cautiously. The remote host deletes the file when accepting the job. However, the remote host does not guarantee that it will print or execute the job. (That is, the remote printer might fail, someone could delete jobs from the queue, or you might not have access to the queue). The remote host does not delete the file if the remote queue does not accept the job.

-t

File contains output from UNIX `troff` formatting commands. (Do not append any characters onto the `t` of the option or it can be interpreted as an argument to the uppercase `-T` option. See the Note.)

-"T"*title* (or **-t***title*)

Prints a title on the first page of output. Use this option only when you use the -p option to format a file. (See the previous note for details on syntax.)

-v

File is in Sun raster format.

-w*number*

Width of the output pages in characters. (Do not leave a space between the `-w` and the *number*.)

-z*number*

Length of the output pages in lines. (Do not leave a space between the `-z` and the *number*.)

-1*string*
-2*string*
-3*string*
-4*string*

The options name UNIX font files and work the same as they do in UNIX. (Do not leave a space between the number and the string.)

Use these options only with the `-d, -n,` and `-t` options.

-#*number*

Prints multiple copies, where *number* is the number of copies you want of each file. (Do not leave a space between the `-#` and the *number*.)

Note! LPR does not support the UNIX `lpr` options `-s` and `-q`.

Some LPD servers that reside on non-UNIX hosts (such as the one provided by TCPware) do not support the following UNIX `lpr` options: `-p, -t, -n, -d, -g, -v, -c, -i, -w, -z, -1, -2, -3,` and `-4`.

Examples

1 This command prints the file MEMO.TXT on the default remote printer:

```
LPR MEMO.TXT
```

2 Each of these commands send the file MEMO.TXT to the remote printer specified by the logical `drp02`:

```
LPR -"P"drp02 MEMO.TXT
lpr -pdrp02 memo.txt
```

3 Each of these commands send the file MEMO.TXT to the remote printer `lp` at host `daisy`:

```
LPR -"P"lp@daisy MEMO.TXT
lpr -plp@daisy memo.txt
```

4 Each of these commands specify mymemos as the job name on the burst page, and send MEMO1.TXT, MEMO2.TXT, and MEMO3.TXT to the default remote printer:

```
LPR -"J"mymemos MEMO1.TXT MEMO2.TXT MEMO3.TXT
lpr -jmymemos memo1.txt memo1.txt memo2.txt memo3.txt
```

5 This command sends three copies of the MEMO.TXT and LETTER.TXT files to the default remote printer:

```
LPR -#3 MEMO.TXT LETTER.TXT
```

6 This command:

```
LPR -t -h -w72 MEMO.LIS
```

Indicates that:

- The file contains UNIX troff formatting code.
  - The burst page should not be printed.
  - The width of the output should be 72 characters.
- The MEMO.LIS file is sent to the default remote printer.

LPRM

Removes one or more jobs from a remote print queue.

You can remove jobs from remote queues in these situations only:

- The jobs were submitted from your local host
- Your local host has direct access to the remote host. The following files define this access:
  - TCPWARE:LPD_USERS.DAT (the LPD Access File) for TCPware hosts
  - `etc/hosts.lpd` **or** `/etc/hosts.equiv` for UNIX hosts

When removing remote jobs from an OpenVMS host, use the LPRM command instead of the OpenVMS DELETE/ENTRY command. LPRM removes files from the TCPWARE_LPD_SPOOL directory, whereas DELETE/ENTRY does not.

The LPRM command displays a message only when it removes a job or encounters an error. If it does not delete a job (such as when the queue is empty), a message does not appear.

You can specify up to 50 jobs and 50 usernames on one LPRM command line.

Format

LPRM *[-*"P"*printer] [job-number ...] [username ...] ["-"]*

TCPware converts all uppercase letters to lowercase unless you enclose them in quotation marks ("").

If you omit a *job-number* or *username* and you own the job that is currently active, TCPware removes the job.

Parameters

job-number

Specifies which job you want removed from the remote queue. If you omit this parameter, TCPware removes the currently active job.

Use the LPQ command to display the *job-number* of a job.

username

Specifies the owner of the jobs you want removed from the remote queue. TCPware removes all jobs the specified user owns.

You can remove jobs that you do not own from a remote queue only under these conditions:

- The remote host is an OpenVMS host
- Your local account is mapped to an OpenVMS username that has OPER privilege on the remote host

  Use the LPQ command to display the *usernames* for all jobs.

Options

-"P"*printer@host*
-"P"*logical-name*

Specifies a remote printer. If you omit this option, TCPware removes the job from the queue the TCPWARE_LPR_PRINTER logical defines.

"-"

If you have OpenVMS OPER privileges on the local host, this option removes all jobs your local host submitted to the remote queue. Otherwise, it removes only your jobs.

Place quotation marks (" ") around this option if it is the last character on the command line because OpenVMS treats trailing hyphens as continuation line indicators.

Do not enter this option when you enter the *job-number* or *username* parameters.

Examples

1 This command removes your currently active job from the default remote print queue:

```
LPRM
```

2 This command removes all jobs that belong to user smith from the lp queue on host daisy:

```
LPRM -"P"lp@daisy SMITH
```

3 This command removes jobs 489, 490, and 495 from the default remote print queue. You can issue this command if you own these jobs, or you have OpenVMS OPER privilege on the remote host:

```
LPRM 489 490 495
```

4 If you have OpenVMS OPER privilege on the local host, this command removes all jobs from the default remote print queue. If you do not have this privilege, this command removes only the jobs you own.

```
LPRM "-"
```

PRINT

Queues jobs for printing on a local or remote printer.

Useful for sending a print job to a printer attached to a terminal server.

For details on Terminal Server Print Services implementation, see the /QUEUE qualifier.

The OpenVMS process that controls OpenVMS queues determines the remote printer by checking the following items in this order:

1 The TCPWARE_LPR_*qname*_PRINTER system logical

2 The /PARAMETERS qualifier

3 The TCPWARE_LPR_*qname*_PRINTER_DEFAULT system logical

Information in this section applies only to using the TCPware for OpenVMS PRINT command with LPS and Terminal Server Print Services.

HP OpenVMS documentation provides complete information on the PRINT command.

Format

PRINT *file-spec[, file-spec...]*

Parameter

file-spec

Specifies the file (or files if separated by commas) you want printed.

Qualifiers

/COPIES=*n*

Prints multiple copies of output, where n is the number of copies.

If you place this qualifier immediately after the PRINT command, each file listed in the command string prints *n* times. (The same effect occurs when you use the **`-#number`** option with the LPR command.) If you place this qualifier after a file specification, only that file prints *n* times.

/FORM=*form-name*

Specifies the name or number of the form you want associated with the print job. If omitted, the default form for the execution queues with the job.

Forms have attributes such as print image width and length or paper stock. To see which forms are defined for your system, use the SHOW QUEUE/FULL command.

/NAME=*job-name*

Names the job. If you do not use this qualifier, the job name is the name and extension of the first file in the job. This name displays on the screen when you use the LPQ command to request queue information, and on the flag page.

This qualifier is equivalent to the **`"-J"`** option used with the LPR command.

/NOFLAG

Suppresses printing of the burst page.

This qualifier is equivalent to the **`"-h"`** option used with the LPR command.

/NOTE=*string*

Names the job classification you want used on the burst page. If you omit this qualifier, the job classification is the domain name of the local host.

/PARAMETERS=*(parameter-1[,...,parameter-8])*

Allows you to specify UNIX LPR command options that do not have OpenVMS equivalents. If you enter only *parameter-1*, you can omit the parentheses. You can enter up to eight parameters:

parameter-1	sends jobs to a specific remote printer. Enter either a system logical name or *printer@host*. This parameter overrides the printer defined by the TCPWARE_LPR_*qname*_PRINTER_DEFAULT logical. If you choose to use the default printer and want to enter subsequent parameters in the same command line, you must enter double quotation marks (`""`) in place of *parameter-1*.
parameter-2 through **parameter-8**	specify the following LPR UNIX options: `c, d, g, i, l, m, n, o, p, t, T, v, x, w, z, 1, 2, 3, 4`. You can use leading hyphens, but they are not required. Enclose the option in quotation marks (for example, `"t"`). (The `f` option is unnecessary; the OpenVMS process that controls OpenVMS print queues automatically specifies this filter for FORTRAN carriage control files.)

Each parameter can include more than one option. However, you must enclose all options within the same set of quotation marks (for example, `"m g","i d"`)

Note! Some LPD servers that reside on non-UNIX hosts (such as the one provided by TCPware) do not support the following LPR UNIX options: `-p, -t, -n, -d, -g, -v, -c, -i, -w, -z, -1, -2, -3, -4`.

/PASSALL/NOPAGE

Uses a transparent filter so that you can send data to the printer unchanged. Note that this command qualifier DOES NOT convert the file to STREAM-LF format. This qualifier is equivalent to the `-l` option used with the LPR command.

When using *-to-OpenVMS printing, the /PASSALL qualifier prints text files without carriage returns (CRs). Use this option mainly with BINARY data or a file that contains all of the characters (including CRs) that you want sent to the printer.

If you use LPS and issue the PRINT command, the printing process ignores the /BURST, /CHARACTERISTIC, /HEADER, /PAGES, /SETUP, /SPACE, and /TRAILER OpenVMS PRINT qualifiers. All other OpenVMS PRINT qualifiers work the same as they normally do with OpenVMS.

/QUEUE=qname

Specifies a print queue that can send the job to a local or remote printer. If you omit this parameter with Line Printer Services, the job goes to the SYS$PRINT queue.

The /QUEUE parameter is necessary when generating a print request on a remote printer attached to a terminal server (when using the Terminal Server Print Services). Once the server initializes and starts the print queue for a terminal server print job, you can generate a print request on the terminal printer as follows:

```
PRINT/QUEUE=qname filename
```

The *qname* parameter is the name of the print queue, and the *filename* parameter specifies the data file or files you want used. The standard OpenVMS qualifiers are available.

VMSLPR Symbiont

By default the VMSLPR symbiont generates a flag page locally using the VMS print symbiont and suppresses the banner page generated by the LPD server. You can make the VMSLPR symbiont request a banner page from the LPD server on a specific queue by defining the logical:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_VMSLPRSMB_<queue-name>_REMOTE_BANNER "TRUE"
```

To enable this functionality on all VMSLPR symbionts, define the logical:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_VMSLPRSMB_REMOTE_BANNER "TRUE"
```

The following logical has been added to the VMSLPR symbiont allowing you to define the number of characters you want removed from the end of a print job.

```
$ DEFINE/SYS/EXEC TCPWARE_VMSLPRSMB_<queue-name>_TRIMTAIL #
```

– **#** is a numeric value indicating the number of characters to remove from the end of each print job. If not specified, the default value is 2.

Examples

1 This command prints the file MEMO.TXT on the remote default printer:

```
PRINT/QUEUE=LPR$PRINT MEMO.TXT
```

2 This command sends the file MEMO.TXT to the SYS$PRINT queue, which is usually a local printer:

```
PRINT MEMO.TXT
```

3 This command prints the file MEMO.TXT on the lp printer at host DAISY. You can enter this command only if you did not define the system logical TCPWARE_LPR_LPR$PRINT_PRINTER.

```
PRINT/QUEUE=LPR$PRINT /PARAMETERS="lp@DAISY" MEMO.TXT
```

4 This command:

```
PRINT/QUEUE=LPR$PRINT /PARAMETERS=("lp@DAISY","m","t") MEMO.TXT
```

Is identical to the previous example, with these additions:

– The user who issued the command receives a mail message when the job completes.

– The file contains UNIX troff commands.

5 This command is identical to the previous example except that *parameter-1* is omitted:

```
PRINT/QUEUE=LPR$PRINT /PARAMETERS=("","m","t") MEMO.TXT
```

The result is that the file MEMO.TXT goes to the printer defined by the TCPWARE_LPR_LPR$PRINT_PRINTER_DEFAULT logical.

PRINT Command Options

Print command options are specified using the OpenVMS standard /PARAMETERS qualifier. The list of options is enclosed in parenthesis. For example,

```
$ PRINT /QUEUE=IPP_PRINTER_1
/PARAMETER=(COPIES=3, ORIENTATION=LANDSCAPE) FILE.TXT
```

These options are not case sensitive. The underscores in the option names are optional. Each may be abbreviated as long as the result is not ambiguous.

The available print command options are:

PRINTER=*printer_uri*

Specifies the target printer when the queue default is not desired, or when there is no queue default. The printer URI specified must match at least one of the defined printer_uri's for the print queue.

Wildcards cannot be used in the printer URI.

COPIES=*number*

Specifies the number of copies of each document to print. The default value is 1.

SIDES=*keyword*

Specifies how the printing is to be placed on the paper. The *keyword* must be one of the following:

- ONE-SIDED or 1sided: prints each consecutive page upon one side of consecutive media sheets.
- TWO-SIDED-LONG-EDGE or two-long-edge or 2long_side: prints each consecutive pair of pages upon the front and back sides of consecutive media sheets, with the orientation of each pair of pages on the long edge. This positioning is called “duplex” or “head-to-head” also.
- TWO-SIDED-SHORT-EDGE or two-short-edge or 2short_side: prints each consecutive pair of pages upon front and back sides of consecutive media sheets, with the orientation of each pair of print-stream pages on the short edge. This positioning is called “tumble” or “head-to-toe” also.

ORIENTATION=*keyword*

Specifies the page orientation. The *keyword* must be one of:

- PORTRAIT
- REVERSE_PORTRAIT
- LANDSCAPE
- REVERSE_LANDSCAPE

These can be abbreviated to any non-ambiguous prefix. Case is ignored.

[NO]FLAG

Requests, or suppresses, the printing of an IPP flag page for the job. The printer may, or may not, respond to this request. The exact format of this flag page is up to the IPP Server (printer) implementation.

NUMBER_UP=*number*

Specifies the number of page images to be placed on each side of each sheet of paper. The number must be an integer that is acceptable to the IPP server. If the number specified is not a value supported by the server, the job aborts.

DOCUMENT_FORMAT=*MIME-media-type* or **DOCUMENT_FORMAT=***printer_default*****

Specifies the document format of the files in the job, or specifies use of the printer's built-in default. The default for this qualifier is the default for the queue. Also, if the queue configuration does not specify a default document format, the hard coded default is "text/plain".

JOB_PRIORITY=*integer*

Specifies the priority of the print job at the IPP server (not to be confused with the OpenVMS queue priority). 1 is the lowest, 100 is the highest.

FINISHINGS=“*keyword[,keyword]...”*

Specifies finishing operations to be performed on the printed documents. May or may not be supported by a given IPP server. Any or all of the four available finishings may be specified. Case is ignored.

- BIND
- COVER
- PUNCH
- STAPLE

MULTIPLE_DOCUMENT_HANDLING=*keyword*

Specifies how you want the printer to print your job. The *keyword* is one of the following:

- Single_Document or 1Document
- Separate_Documents_Uncollated_Copies or UncollatedSeparate
- Separate_Documents_Collated_Copies or CollatedSeparate
- Single_Document_New_Sheet or NewSheet

Case is ignored. See /MULTIPLE_DOCUMENT_HANDLING_DEFAULT=keyword in Chapter 15 of the *TCPware Management Guide* for information on single document, separate-documents-uncollated-copies, separate-documents-collated-copies, and single-document-new-sheet handling.

PAGE_RANGES="*range[,range]...*"

Specifies the page numbers to print. *range* is either a single integer page number, or a pair of page numbers, separated by a hyphen. Multiple range specifications are separated by commas and enclosed in double quotes.

For example:

```
$ PRINT/QUEUE=IPP_QUEUE/PARAM=(PAGE_RANGES="1,3-6,  9, 10,  12-14") FILE.TXT
```

Note that embedded spaces are allowed, and ignored. The example specifies the pages: 1, 3, 4, 5, 6, 9, 10, 12, 13, and 14.

MEDIA=*name*

This attribute identifies the medium that the Printer uses for all pages of the Job.

The values for "media" include medium-names, medium-sizes, input-trays and electronic forms. See your printer documentation for details concerning what values are supported for your printer.

Standard keyword values are taken from ISO DPA and the Printer MIB and are listed in section 14 of RFC 2566. Some servers may support definition of locally created names as well.

See Standard Media Names and Input Tray Names for the standard media names.

QUALITY=*keyword*

Specifies the quality of the printed material. Case is ignored. The keyword choices are:

- DRAFT
- HIGH

- NORMAL

Table 5-1 contains examples of standard names. These names include, but are not limited to the following:

Table 5-1 Standard Media Names

Name	**Description**
default	The default medium for the output device
iso-a4-white	Specifies the ISO A4 white medium
iso-a4-colored	Specifies the ISO A4 colored medium
iso-a4-transparent	Specifies the ISO A4 transparent medium
na-letter-white	Specifies the North American letter white medium
na-letter-colored	Specifies the North American letter colored medium
na-letter-transparent	Specifies the North American letter transparent medium
na-legal-white	Specifies the North American legal white medium
na-legal-colored	Specifies the North American legal colored medium
na-9x12-envelope	Specifies the North American 9x12 envelope medium
monarch-envelope	Specifies the Monarch envelope
na-number-10-envelope	Specifies the North American number 10 business envelope medium
na-7x9-envelope	Specifies the North American 7x9 inch envelope
na-9x11-envelope	Specifies the North American 9x11 inch envelope
na-10x14-envelope	Specifies the North American 10x14 inch envelope
na-number-9-envelope	Specifies the North American number 9 business envelope
na-6x9-envelope	Specifies the North American 6x9 inch envelope
na-10x15-envelope	Specifies the North American 10x15 inch envelope

executive-white	Specifies the white executive medium
folio-white	Specifies the folio white medium
invoice-white	Specifies the white invoice medium
ledger-white	Specifies the white ledger medium
quarto-white	Specified the white quarto medium
iso-a0-white	Specifies the ISO A0 white medium
iso-a1-white	Specifies the ISO A1 white medium
a	Specifies the engineering A size medium
b	Specifies the engineering B size medium
c	Specifies the engineering C size medium
d	Specifies the engineering D size medium
e	Specifies the engineering E size medium

The following standard values are defined for input-trays:

Table 5-2 Input Tray Names

Name	**Description**
top	The top input tray in the printer.
middle	The middle input tray in the printer.
bottom	The bottom input tray in the printer.
envelope	The envelope input tray in the printer.
manual	The manual feed input tray in the printer.
large-capacity	The large capacity input tray in the printer.
main	The main input tray

side	The side input tray

Submitting Jobs to IPP Symbiont Print Queues

This section describes how to submit jobs to the IPP symbiont print queues.

Printing a Single Text File to an IPP Queue

Print the file FOO.TXT to the IPRINTER (default destination printer) set up in the prior examples:

```
$ PRINT/QUEUE=IPRINTER_QUEUE foo.txt
```

Specifying the Destination Printer on the Print Command

Print a single text file to a non-default printer on a queue with a wild carded printer URL:

```
$ PRINT /QUEUE=iprinter_queue -
_$ /PARAM=(printer="ipp://another.mynet.com/ipp/port1") foo.txt
```

Note! The above will fail unless the queue specifies *another.mynet.com* as a legal URL, either explicitly or by using wildcards.

Using Other Print Qualifiers

Print a text file to a default printer on a queue but specify the document format and additional copies:

```
$ PRINT /QUEUE=iprinter_queue-_$
_$ /PARAM=(document="text/plain" copies=3) foo.txt
```

TCPWARE IPP SHOW Command

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. When called from a DCL script or other program, it can be used to gather information about a number of printers, or used to match printer capabilities with the needs of a given print job.

For detailed information on the IPP SHOW command, see Chapter 15 of the *TCPware Management Guide*.

Chapter 6
RCD and RMT: Remote CD-ROMs and Tapes

Introduction

The Remote Magnetic Tape (RMT) Client and Remote Compact Disc (RCD) Client provide access to tape drives and CD-ROM drives, respectively, on remote TCP/IP systems. This chapter describes how to set up RMT and RCD on your OpenVMS system so that you can use the commands typically associated with tape and CD-ROM drives, such as BACKUP, MOUNT, COPY, and EXCHANGE.

RMT Client and RCD Client

To use a remote tape or CD-ROM, you must first "connect" to the server system with the RMTSETUP command, which creates a pseudodevice. You can then use OpenVMS commands such as BACKUP, MOUNT, COPY, and EXCHANGE. These are the same commands issued directly to the physical tape or CD-ROM device on the server. When you conclude the activity, you can discard the pseudodevice using the DEALLOCATE command.

Note that not all tape drives or CD-ROM drives can fully support use of the RMT Client and RCD Client. For example, quarter-inch tape drives on UNIX systems typically support only fixed-length, 512-byte records. You cannot use these tapes with the OpenVMS COPY or BACKUP commands because the latter require variable-length records.

An attempt to perform an unsupported operation to a remote device results in a `%SYSTEM-E-UNSUPPORTED` error message.

Troubleshooting

You can lose the TCP/IP connection between the RMT or RCD client and server if:

- An RMT or RCD server receives a command that it does not recognize. Rather than returning an error message, it simply closes the connection.
- A bug in an RMT or RCD server causes it to crash.
- If the RMT or RCD server (or its system) crashes or is shut down.

In these situations, the RMT Client or RCD Client detects the loss of the TCP/IP connection and returns the following error message for all subsequent commands:

```
%SYSTEM-F-LINKABORT, network partner aborted logical link
```

The only alternative at this point is to deallocate the existing device and reconnect to the server (when it becomes available) by running RMTSETUP again.

RMTSETUP

Configures an RMT or RCD pseudodevice, `_RMTn:` or `_RCDn:`, respectively, on your local OpenVMS system. In this way you can perform functions on remote magnetic tape or CD-ROM drives connected to an RMT or RCD server. The remote RMT or RCD server must support the `rmt` protocol.

Connecting to a remote CD-ROM drive requires the /CD qualifier. You can connect to the remote host with a different username by specifying the optional /USERNAME qualifier on the command line.

Format

RMTSETUP *host remote-device [logical]*

Parameters

host

Name or internet address of the host on which the remote tape or CD-ROM drive resides. This host must have an RMT server available.

remote-device

Name of the remote tape device (such as `MKB500:`) or CD-ROM device (such as `DKA200:`) on the RMT server. If sending the device and any server options to a non-TCPware server, you must enclose this information in double quotes, such as **`"/dev/rst0"`** for a UNIX server with "read-only" privileges.

logical

Optional OpenVMS logical assigned to the newly created pseudodevice. If omitted, RMTSETUP uses the logical name **`TCPWARE_TAPE`** for tapes and **`TCPWARE_DISK`** for disks.

Qualifiers

Not all RMT servers support the following RMT Client qualifiers as options or qualifiers. For UNIX servers, for example, you must include options as part of *remote-device* as a quoted string. For example, `"/dev/mt0"` is a stream device and `"/dev/rmt0"` is a non-stream device. With a TCPware RMT server, where *remote-device* is not a quoted string, the Client qualifiers that are also server qualifiers are sent to the server.

/ASSIST (default)
/NOASSIST

Action to take when the device cannot mount on the remote system. With /ASSIST, operator messages appear on the remote system indicating corrective action to take (if supported). With /NOASSIST, only a local message appears. Not allowed when used with /CD.

Note! The BACKUP command's /ASSIST and /NOASSIST qualifiers further direct messages to the local operator and user, respectively.

/BLOCKSIZE=*size*

Default block size of the remote tape device. Not allowed when used with /CD.

/CD

Indicates that the remote device is a CD-ROM device.

/COMMENT=*"string"*

Used with the /ASSIST qualifier to send a message to the remote operator when a mount operation fails. Not allowed when used with /CD.

/DENSITY=*density*

Density, in bits per inch, at which to write the remote tape. Not allowed when used with /CD.

/LOG
/NOLOG (default)

Displays log information during RMTSETUP execution.

/MOUNT (default)
/NOMOUNT

/MOUNT allows the user exclusive access to the device. /NOMOUNT disables exclusive access to the device. /NOMOUNT also prevents a remote tape from rewinding when deallocating the pseudodevice on the client. Not allowed when used with /CD.

You cannot combine /NOMOUNT with /ASSIST, /BLOCKSIZE, /COMMENT, or /DENSITY.

Use /NOMOUNT carefully since it allows multiple users access to the same device.

/PASSWORD*[=password]*
/NOPASSWORD

/PASSWORD sets the password to access the remote system and causes the RMT server to use the `rexec` rather than the `rshell` service. The *password* is converted to lowercase unless you enclose it in quotes. /NOPASSWORD uses the `rexec` service with a blank password. Without either qualifier, access to the remote tape device is controlled through the `TCPWARE:HOST.EQUIV` and `SYS$LOGIN:.RHOSTS` files. Use together with /USERNAME.

Using the *password* value can pose a security risk. Also, using a null password for which you have to be prompted can cause an error in a command procedure.

/REWIND (default)
/NOREWIND

/REWIND rewinds a tape before its initial use. /NOREWIND causes the tape to stay in an arbitrary position after running RMTSETUP. Not allowed when used with /CD.

/STREAM
/NOSTREAM (default)

A tape is normally written as a series of records. /STREAM ignores record boundaries and returns data read from the tape as a stream of bytes (the UNIX model). Not allowed when used with /CD.

Most OpenVMS utilities expect tape drives to operate in non-stream mode, so take care in overriding the /NOSTREAM default.

/TRUNCATE_USERNAME*[=length]*

Truncates the username sent to the RMT server to the specified *length* to accommodate requirements of some non-OpenVMS systems. The default *length* is 8.

/UNLOAD (default)
/NOUNLOAD

/UNLOAD unloads the remote device when deallocating the local RMT pseudodevice. /NOUNLOAD disables this. Note that a DCL MOUNT or DISMOUNT with /UNLOAD or /NOUNLOAD overrides the RMTSETUP /UNLOAD or /NOUNLOAD. Not allowed when used with /CD.

/USERNAME=*username*

Username for access to the remote system. If omitted, the username of the client process is sent to the server (subject to truncation by /TRUNCATE_USERNAME). *Username* is converted to lowercase unless you enclose it in quotes. Use together with /PASSWORD.

/WRITE (default)
/**NOWRITE**

Writing to the remote tape is usually enabled. /NOWRITE is a precautionary measure to prevent a remote tape from being written. Not allowed when used with /CD.

Examples

1 This example uses tape drive `MKB500:` on remote OpenVMS system IRIS to back up all the TCPWARE data files that start with `SM`. The tape is left loaded in the drive after its use (/NOUNLOAD). `MYTAPE` is the logical name for the `_RMT9:` device created.

```
$ RMTSETUP IRIS MKB500: MYTAPE /NOUNLOAD /LOG
Connecting to RMT server on host IRIS through port 514 (rsh)
Opening MKB500:/NOSTREAM/NOUNLOAD
_RMT9: created
$ BACKUP /LOG TCPWARE:SM*.DAT MYTAPE:TCPWARE.BCK /SAVE_SET
%MOUNT-I-MOUNTED, TEST1 mounted on _RMT9:
%BACKUP-S-COPIED, copied SYS$SPECIFIC:[TCPWARE]SM.DAT;1
%BACKUP-S-COPIED, copied SYS$SPECIFIC:[TCPWARE]SM_BAK.DAT;1
$ DISMOUNT MYTAPE /NOUNLOAD
$ DEALLOCATE MYTAPE
```

2 This example requests access to tape drive `/dev/rst0` on a remote UNIX system, using a username and password. The `initialize` command was unrecognized by the tape drive on the UNIX system and rejected. The `tar` utility examines the contents of the tape, which was written from the UNIX system. (`tar` is available over the network and is an alternative to the EXCHANGE utility).

```
$ rmtsetup sigma.nene.com "/dev/rst0" -
_$ /username=system /password
Password for root on host SIGMA.NENE.COM:
$ initialize tcpware_tape test
%INIT-F-UNSUPPORTED, unsupported operation or function
$ mount /foreign /record_size=512 tcpware_tape
$ tar -ftv tcpware_tape
644   4069 Jun 1 16:29:21 2001 /etc/hosts
End of Tar file found.
Do you wish to move past the EOF mark (y/n)? n
$ dismount tcpware_tape
$ deallocate tcpware_tape
```

3 This example requests access to CD-ROM drive `DKA100:` on remote host `roman`, mounts the CD-ROM using `MY_CD` as the logical name, and requests a directory listing:

```
$ rmtsetup /cd /log roman dka100: my_cd
Connecting to RCD server on host ROMAN through port 514 (rsh)
Opening DKA100:
_RCD1: created
$ mount my_cd /override=id
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-MOUNTED, OPENVMS062   mounted on _ALTARF$RCD1:
$ dir my_cd:[0,0]
```

Chapter 7 RCP: Copying Files

Introduction

The Remote Copy Program (RCP) is a command you can use to copy files between your local OpenVMS host and a remote host. TCPware provides RCP as part of the FTP-OpenVMS product.

For the FTP utility commands, see Chapter 3, *FTP: Transferring Files*. Use the RCP command to copy remote files. You can copy files:

- From a remote host to your host
- From your host to a remote host
- From one remote host to another remote host (a "third-party" copy)

CAUTION! If you are using RCP with Kerberos version 4 authentication in a third-party copy, only the first connection uses Kerberos. The second connection uses standard authentication, in which case the username and password pass through the network as clear text.

Before you use RCP, your system manager must install and configure the TCPware FTP-OpenVMS product and enable the shell service during TCPware R Services configuration.

Also, make sure your host or username is registered in the remote system's `~/.rhosts` (if UNIX) or SYS$LOGIN:.RHOSTS file (if OpenVMS). To use Kerberos version 4 authentication with the remote host, be sure your username and Kerberos realm are in the remote host's `~/.klogin` file (if UNIX) or SYS$LOGIN:.KLOGIN file (if OpenVMS).

To use Kerberos version 4 authentication, your system manager must enable the `kshell` service during TCPware's Kerberos Services configuration. If you request Kerberos authentication, RCP tests for it first. If the test fails, RCP uses standard authentication instead.

With Kerberos V4 authentication, you can specify the Kerberos realm using the /REALM qualifier. If omitted, the TCPWARE_KERBV4_REALM logical value determines the realm.

If you need to preserve case for any of the command elements, enclose each in quotes, since RSH lowercases unquoted text strings. Include a pair of quotes for each redirection of the command. If you are redirecting a command through one remote host to have it executed on a third, each host in turn strips off a pair of quotes after interpreting the command. In this case, you may need three pairs of quotes around the command element in order to preserve case.

RCP

Copies files between the local and remote host, or between two remote systems.

Format

RCP *source destination*

Parameters

source

Source host and pathname information, in the general format:

host:filespec

- *hos*t is the remote host name followed by a colon (:).
- *filespec* is different for UNIX and OpenVMS systems:
  - For UNIX system source hosts, use the absolute pathname (such as **`/etc/user/hosts`**) or the one relative to the user's home directory (**`hosts`**).
  - For OpenVMS source hosts, use the format***[dir]file.typ***, or ***file.typ***, which assumes the current directory.

If you include a *username* or *device*, use the following format:

```
"username@host:device:filespec"
```

If you include a username and want to copy from a remote host, the remote host must include your host (and username) in its host equivalence file. If you do not use the above format, use the /USER, /PASSWORD, and /TRUNCATE qualifiers.

Note! Do not use /USER or /PASSWORD when using DECnet syntax for a source or destination: *`host"username password"::filespec`*

You also cannot use DECnet syntax for both source and destination (as for a remote-to-remote copy) that involves two passwords.

destination

Destination host and pathname information, in the same format as source.

Qualifiers

/**AUTHENTICATION*[=auth-type]***

Determines the authentication method. If **`KERBV4`** (or you omit the value), uses Kerberos v4 authentication. If **`NULL`** (or you omit the qualifier), uses standard authentication.

/**LOG**

Logs the files copied to or from the local system. The default is not to log. Logging only applies to the first remote host transaction in a third-party copy.

/VMS[={MULTINET | TCPWARE (default)}]
/**NOVMS**

If /VMS is omitted, RCP by default attempts a TCPware style VMS mode transfer. This retains VMS file attributes across copies. Use /VMS=MULTINET to do a transfer involving a MultiNet machine. Use /NOVMS only if you get the error `%DCL-W-IVKEYW, unrecognized keyword - check validity and spelling` with the RCP command. /NOVMS disables maintaining VMS file attributes during a third-party copy.

/PASSWORD=*remote-password*

Password for the remote account. Use with the /USER qualifier. Do not use with DECnet source or destination syntax.

CAUTION! The password is sent across the network as plain text.

/PRESERVE

-p

Preserves the file protection mode and modification date during a copy. /PRESERVE and **`-p`** are equivalent.

/REALM=*realm*

Assigns the name of the Kerberos realm. Use if the Kerberos Server resides in a different realm than the local host. Use with the /AUTHENTICATION=KERBV4 qualifier and value. The RCP client converts *realm* to lowercase unless you enclose it in quotes.

/RECURSIVE

-r

Recursively copies each subtree rooted at the directory you specify in the UNIX system *filespec*. This makes it possible to copy entire UNIX system directories and their files. In OpenVMS, specify **[*dir...*]** (with the three trailing dots) in the *filespec* instead of using /RECURSIVE. **`-r`** is the UNIX system equivalent.

/USER=*remote-username*

User on the remote host. Use only if the remote host's ~/.hosts or .HOSTS file does not include your local host name or username. If necessary, truncate username to the required number of characters using the /TRUNCATE qualifier. Converted to lowercase if not enclosed in quotes. Do not use with DECnet file syntax.

/TRUNCATE*[=n]*

Truncates the username to the specified n number of characters, since some UNIX systems restrict the length of usernames. If you omit *n*, the default is eight characters.

Examples

1 This command copies a remote UNIX system source file in its home directory to a local host file of the same name in the current directory. The copy preserves the source file's protection mode and modification date.

```
rcp /preserve unixhost:src_file []
```

2 This command copies the complete remote UNIX system directory tree ~/src_dir to the local subdirectory DST_DIR, while logging the copy of each file:

```
rcp /recursive/log unixhost:src_dir [.dst_dir...]
```

3 The first of these two commands only copies the .src_dir subdirectory to a UNIX system. The second command copies the whole subtree.

```
rcp /recursive [.src_dir] unixhost:dst_dir
rcp [.src_dir...] unixhost:dst_dir
```

4 This command copies the complete local subdirectory tree SRC_DIR to a remote OpenVMS host's destination directory while preserving the directory hierarchy:

```
rcp [src_dir...] vmshost:[dst_dir...]
```

5 This command copies all files under the local SRC_DIR directory to a remote OpenVMS host's destination directory. This does not preserve the copied directory's hierarchy:

```
rcp [src_dir...] vmshost:[dst_dir]
```

6 This command copies all directories and files under the local SRC_DIR directory to a remote OpenVMS host user's login directory on the DKA300: device (use the double quotes):

```
rcp [src_dir...] "vmshost:dka300:[login...]"
```

7 This command copies the local SRC_FILE on device DKA100: to `dst_file` on a remote host. Double quotes are needed to specify a device name. The /NOVMS qualifier allows RCP to copy compatibly to an OpenVMS host running HP TCP/IP Services for OpenVMS (UCX).

```
rcp /novms ":dka100:[src_dir]src_file" ucx_host:dst_file
```

8 This command copies the local SRC_FILE to `~some/dst_file` if the remote host is a UNIX system, or [`some`-*login-directory*]DST_FILE if the remote host is OpenVMS. (RCP truncates the `someone` username to `some`.) In this case, the remote host does not have a host equivalence file entry for the local host, requiring /USER and /PASSWORD.

```
rcp /user=someone/pass=password/truncate=4 src_file host:dst_file
```

9 Each command copies a UNIX system file to the local host's current directory. The `-p` switch in the first command precludes having to use double quotes around the UNIX system file specification. The second command is the equivalent without the `-p` switch.

```
rcp -p unixhost:/usr/users/src_file []
rcp "unixhost:/usr/users/src_file" []
```

10 This command copies a file from one remote host to another (a "third-party" copy):

```
rcp remotehost1:file1 remotehost2:file2
```

11 This command copies a remote UNIX system source file in its home directory to the DST_FILE filename on the local host under the current directory. Uses Kerberos V4 authentication. The Kerberos Server and its database reside in the realm `daisy.com`.

```
rcp /auth=kerbv4 /realm=daisy.com unixhost:src_dir dst_file
```

12 This command copies all files under the local SRC_DIR directory to a remote OpenVMS host's destination directory while preserving the directory hierarchy. Since the /AUTHENTICATION qualifier appears without a value, Kerberos V4 authenticates the user to the remote UNIX host. Because /REALM is omitted, the TCPWARE_KERBV4_REALM logical value determines the Kerberos realm.

```
rcp /auth [src_dir...] vmshost:[dst_dir...]
```

13 Using the /USER or /PASSWORD qualifier with DECnet syntax is not allowed and returns the error message shown:

```
rcp /user=user1 new.txt flower"user2 password"::new.txt
TCPWARE-E-NOQUAL, /USERNAME qualifier not allowed with DECnet syntax
```

14 Using multiple passwords with DECnet syntax is not allowed and returns the error message shown:

```
rcp tree"user1 pass1"::new.txt flower"user2 pass2"::new.txt
TCPWARE-E-MULTPW, Multiple passwords not supported
```

Chapter 8 RLOGIN: Logging In to a Remote Host

Introduction

RLOGIN is the Berkeley R Command utility you can use to log in to a remote host. RLOGIN provides a functionality similar to TELNET except that RLOGIN follows more of a UNIX format.

This chapter is a basic use summary of the RLOGIN command.

Before you use RLOGIN, be sure your host or username is registered in the remote system's `~/.rhosts` file (if UNIX) or SYS$LOGIN:.RHOSTS file.

See the *Management Guide*, Chapter 16, *Managing R Commands*, for information on host equivalence files.

To use Kerberos version 4 authentication with the remote host, make sure that your username and Kerberos realm are in the remote host's `~/.klogin` file (if UNIX) or SYS$LOGIN:.KLOGIN file (if OpenVMS).

To use Kerberos V4 authentication, your system manager must configure TCPware's Kerberos Services. You must also first get a ticket-granting ticket (TGT) from the Kerberos Server.

See Chapter 4, *Kerberos User Commands*, for details on getting a TGT.

With Kerberos V4 authentication, you can specify the Kerberos realm using the /REALM qualifier. If omitted, the TCPWARE_KERBV4_REALM logical value determines the realm.

RLOGIN first tries to use Kerberos V4 authentication if requested, then falls back to using standard authentication if Kerberos authentication fails.

To close an RLOGIN connection, simply log out of the remote system.

If you are designated by the system administrator as having password authentication using Token Authentication, you need to enter the PASSCODE in addition to the username and password at a separate `PASSCODE`: prompt. Depending on which type of SecurIDW card you were assigned, do one of the following:

- Enter a combination of your personal identification number (PIN) and the tokencode that appears on the card (with no separating space) at the `PASSCODE`: prompt
- Enter your PIN on the PINPAD] card and the resulting tokencode that appears on the card at the `PASSCODE`: prompt.

See Chapter 14, *Token Authentication: Protecting Logins*, for details on obtaining PASSCODEs.

RLOGIN

Logs in to a remote host from your local host without entering a remote username and password. The remote host must provide `login` service (for standard authentication) or the `klogin` service (for Kerberos version 4 authentication).

You can log in to the remote host with a different username by specifying the /USER qualifier.

When RLOGIN starts up, it processes the flow control characters Ctrl/S and Ctrl/Q locally unless the remote host instructs otherwise. RLOGIN passes all other keystrokes directly to the remote process and perform according to conventions established on the remote host.

The special RLOGIN commands in Table 8-1 are available once you start the connection to the remote host. Enter the special RLOGIN commands as the first character on a line.

Table 8-1 Special RLOGIN Commands

Command	**Purpose**
~.	Closes the connection and exits RLOGIN.
~^Z	Spawns a subprocess on the local host and connects SYS$INPUT, SYS$OUTPUT, and SYS$ERROR to that process. When the subprocess logs out, control returns to the remote session.
~~	***Note!*** You cannot spawn with CAPTIVE accounts. Sends a single tilde to the remote system.

Format

RLOGIN *host*

Parameter

host

Name or internet address of the remote host where you want to log in.

Qualifiers

/**AUTHENTICATION*[=auth-type]***

Determines the authentication method. If **`KERBV4`** (or you omit the value), uses Kerberos v4 authentication. If **`NULL`** (or you omit the qualifier), uses standard authentication.

/**EIGHTBIT**

Accepts eight-bit data from the terminal and sends it to the remote system. The default is that only seven-bit data is sent.

/ESCAPE_CHARACTER=*char*

New escape character for issuing special RLOGIN commands. The default escape character is the ~ (tilde) character.

To close your session from your local host, use a period (**.**) as the escape command.

/LOG=*file*

Logs a copy of the output to the specified file. Output continues to be directed to SYS$OUTPUT while it is being recorded in the log file. The default is no logging.

/LOWERCASE (default)
/NOLOWERCASE

/LOWERCASE sends your local username to the remote host in lowercase (the default). /NOLOWERCASE preserves any uppercase characters in the local username.

/REALM=*realm*

Assigns the name of the Kerberos realm. Use if the Kerberos Server resides in a different realm than the local host. Use with the /AUTHENTICATION=KERBV4 qualifier and value. RLOGIN converts *realm* to lowercase unless you enclose it in quotes.

/TERMINAL_SPEED=*baud*

Terminal speed in baud rate. The default is the current speed of your terminal.

/TERMINAL_TYPE=*type*

Resets the current terminal type to the specified *type*. The allowable types you can use to override the current type are **`VT100, VT200, VT300`**, and **`VT400`**.

The remote terminal type is the same as the local terminal type. If the terminal's virtual size (rows, columns, or pixels) changes during the RLOGIN session, RLOGIN provides the remote host with the new information.

/TRUNCATE*[=n]*

Truncates the local OpenVMS username to n number of characters. The *n* value must be greater than zero or the command aborts with an error. The default is eight characters.

If the local username is also the remote username (if you omit the /USER qualifier), TCPware also truncates the remote username to the indicated length. However, it never truncates a remote username specified explicitly with the /USER qualifier.

/USER=*remote-username*

Username on the remote host that is different from the username with which you are currently logged in to the local host. TCPware never truncates an explicitly specified remote username (see the /TRUNCATE qualifier). The *remote-username* is converted to lowercase unless you enclose it in quotes or use the /NOLOWERCASE qualifier.

Examples

1 Each of these equivalent commands opens a connection to host IRIS using standard authentication:

```
RLOGIN IRIS
RLOGIN /AUTH=NULL IRIS
```

2 This command opens a connection to remote host IRIS, using Kerberos version 4 authentication. The Kerberos Server resides in the `daisy.com` realm.

```
RLOGIN /AUTH=KERBV4 /REALM=DAISY.COM IRIS /USER="Smith"
```

The quotes around Smith are necessary because the name contains a mix of upper- and lowercase characters that you would want to preserve in sending the command. Without the quotes, RLOGIN converts the name to lowercase, which then may not match the username on the remote host.

3 This command opens a connection to remote host IRIS, using Kerberos version 4 authentication. Because /REALM is omitted, the TCPWARE_KERBV4_REALM logical value determines the Kerberos realm.

```
RLOGIN/AUTH IRIS
```

Chapter 9
RSH: Issuing Commands on a Remote Host

Introduction

RSH is the Berkeley R Command utility you can use to execute a single command on a remote host without logging in. This chapter is a summary of using the RSH command.

Before you use RSH, make sure your host and/or username is registered in the remote system's `~/.rhosts` file (if UNIX) or SYS$LOGIN:.RHOSTS file (if OpenVMS) . See the *Management Guide*, Chapter 16, *Managing R Commands*, for details on host equivalence files.

To use Kerberos version 4 authentication with the remote host, make sure that your username and Kerberos realm are in the remote host's `~/.klogin` file (if UNIX) or SYS$LOGIN:.KLOGIN file (if OpenVMS).

To use Kerberos v4 authentication, your system manager must configure TCPware's Kerberos Services. You must also first get a ticket-granting ticket (TGT) from the Kerberos Server.

See Chapter 4, *Kerberos User Commands*, for details on getting a TGT.

If you request Kerberos authentication, TCPware tests for it first. If the test fails, standard authentication is used instead.

With Kerberos v4 authentication, you can specify the Kerberos realm using the /REALM qualifier. If omitted, the TCPWARE_KERBV4_REALM logical value determines the realm.

RSH

Executes a single command on a remote host. The remote host must provide command execution service.

When the command completes execution on the remote host, the RSH command exits and closes the connection; you return to your local working environment.

RSH writes any output from the command to SYS$OUTPUT; it writes any error from the command to SYS$ERROR, unless overridden with the /OUTPUT or /ERROR qualifier.

Some servers (such as UNIX servers) send output with only line feeds for screen display. To satisfy OpenVMS screen displays, RSH inserts a carriage return by default before each line feed before sending the output to the terminal. If your screen display requires only a line feed, use the /RAW qualifier to bypass the default.

If you need to preserve case for any of the command elements, enclose each in quotes, since RSH lowercases unquoted text strings. Include a pair of quotes for each redirection of the command. If you are redirecting a command through one remote host to have it executed on a third, each host in turn strips off a pair of quotes after interpreting the command. In this case, you may need three pairs of quotes around the command element in order to preserve case.

Format

RSH *host command*

Parameters

host

Name or internet address of the host you want to execute the command on. Can be a domain-style name or an IP address.

command

Name of the command or command string to execute on the remote host.

Qualifiers

/AUTHENTICATION*[=auth-type]*

Determines the authentication method. If *auth-type* is **KERBV4** (or you omit the value), Kerberos v4 authentication is used. If *auth-type* is **NULL** (or you omit the qualifier), standard authentication is used.

/ERROR=*file*

File or device to which to direct error messages from the remote command. The default is /ERROR=SYS$ERROR. (See also the /SYSERROR qualifier.)

/LOG=*file*

Logs a copy of the output to the specified file. Output continues to be directed to SYS$OUTPUT while it is being recorded in the log file. Not valid with /SYSERROR. The default is no logging.

/OUTPUT=*file*

Output file or device to which to direct output from the command. The default is /OUTPUT=SYS$OUTPUT.

/PASSWORD=*remote-password*

Password for the remote account. Use together with the /USER qualifier. The password is sent across the network as plain text.

/RAW
/NORAW (default)

Prevents an extra carriage return from being inserted for screen display. Specifying /NORAW or omitting the qualifier places a carriage return before a line feed character before the line is written to the terminal.

/REALM=*realm*

Assigns the name of the Kerberos realm. Use if the Kerberos Server resides in a different realm than the local host. Use with the /AUTHENTICATION=KERBV4 qualifier and value. *Realm* is converted to lowercase unless you enclose it in quotes.

If omitted, the Kerberos realm is determined by the TCPWARE_KERBV4_REALM logical value.

/SYSERROR

Same as the /ERROR qualifier except that it sends messages to the NLA0 device.

/TRUNCATE*[=n]*

Truncates the local OpenVMS username to the specified n length. The *n* value must be greater than zero or the command aborts with an error. The default is eight characters.

If the local username is also the remote username (if you omit the /USER qualifier), TCPware also truncates the remote username to the indicated length. However, it never truncates a remote username specified explicitly with the /USER qualifier.

/USER=*remote-username*

Remote host's username that is different from the username with which you are currently logged in to the local host. TCPware never truncates an explicitly specified remote username (see the /TRUNCATE qualifier). *Remote-username* is converted to lowercase unless you enclose it in quotes.

Examples

1 This command opens a connection to host IRIS and displays the name of your current working directory:

```
rsh iris pwd
```

2 This command opens a connection to host IRIS for username "Smith" and displays the name of the working directory for "Smith":

```
rsh iris /user="Smith" pwd
```

The quotes around `Smith` are necessary because the name contains a mixture of upper- and lowercase characters that you would want to preserve in sending the command. Without the quotes, the name converts to lowercase and may not match the username on the remote host.

3 This command opens a connection to host IRIS and displays the name of your working directory in a "raw" state on a terminal that requires only line feeds to display the information:

```
rsh iris /raw pwd
```

4 This command executes a `pwd` command on ROSES as sent through VIOLET.

```
$ rsh violet /user=system /password=plastic -
_$ rsh roses /user=root/password="""TCPware""" pwd
```

The `TCPware` password is triple-quoted to preserve case through the transaction. The system strips off the first pair of quotes and executes `rsh roses /user=root/pass=""TCPware""`. VIOLET strips off the second set of quotes and executes `rsh roses /user=root/pass="TCPware"`. ROSES strips off the third and executes `pwd.` In each case, the password string is interpreted literally.

5 This command uses Kerberos version 4 authentication to open a connection to remote host IRIS. The Kerberos Server resides in the `daisy.com` realm. Also displays the name of your current working directory.

```
rsh /auth=kerbv4 /realm=daisy.com iris pwd
```

6 This command uses Kerberos version 4 authentication to open a connection to remote host IRIS and displays the name of your current working directory. Because /REALM is omitted, the TCPWARE_KERBV4_REALM logical value determines the Kerberos realm

```
rsh /auth iris pwd
```

Chapter 10 Sending and Receiving Electronic Mail

This chapter describes how to use OpenVMS MAIL and ALL-IN-1 Mail with TCPware and covers the following major topics:

- Using OpenVMS mail across the network
- Using mail under ALL-IN-1 across the network

Using OpenVMS Mail across the Network

TCPware enhances OpenVMS Mail so you can send and receive mail across the network.

Specifying Addresses

When you use OpenVMS Mail to send mail to a host outside your VMScluster, the message is sent via SMTP (Simple Mail Transfer Protocol). For this reason, you must specify the address so that SMTP accepts the mail correctly. The format for the address is:

```
To: SMTP%"recipient@destination"
```

The string SMTP and the destination system name are not case-sensitive; that is, you can type them in either uppercase or lowercase letters. The destination recipient specification may be case- sensitive, however, depending on the destination system's software. On some UNIX systems, ROOT and root specify two different user names (and hence different electronic mail addresses).

If the address contains a quote, enter the address with either \' or \s as shown in the following example formats:

```
To: SMTP%"\'recipient@destination"
```

or

```
To: SMTP%"\srecipient@destination"
```

If the address is on a local DECnet network, use this format:

```
To: SMTP%nodename::username
```

If the address is on a remote DECnet network, you may use this format:

```
To: SMTP%"'nodename::username'@destination"
```

Note! TCPware assumes that an address containing a double colon (::) is a DECnet address. If an address contains a double colon and is not a DECnet address, SMTP does not handle it correctly.

If you know the recipient's IP address, but not the host name (or if the host name is not registered in the Domain Name System), specify the recipient address as follows:

```
To: smtp%"recipient@[aa.bb.cc.dd]"
```

Where *aa.bb.cc.dd* is the destination system's IP address in dotted-decimal form. You must specify the IP address in square brackets.

The OpenVMS Mail utility also allows you to specify an addressee on the command line:

```
$ MAIL filename addressee
```

To use this form of the command with TCPware, you must enclose the address in quotes (and you must double all existing quotes), as follows:

```
$ MAIL filename "smtp%""recipient@destination"""
```

The following example shows the user sending mail using the OpenVMS MAIL utility to a user named John Smith with a user name of "johns" on system SALES.FLOWERS.COM.

```
$ MAIL
MAIL>SEND
To:     SMTP%"johns@sales.flowers.com"
Subj:   This is a test message.
Enter your message below. Press Ctrl/Z when complete, or
Ctrl/C to quit:
Hi John, this is a test of the TCPware extension to the VMS MAIL utility.
Ctrl/Z
MAIL>EXIT
$
```

You receive network mail as you would all other mail in the VMS MAIL utility. The following example shows the user "WHORFIN" reading an SMTP mail message sent by the user "johns."

```
$
New mail on node KAOS from SMTP%"johns@sales.flowers.com" "John Smith"
$ MAIL
You have 1 new message.
MAIL>READ/NEW
#1          03-23-2014 10:05:40.79
From:   SMTP%"johns@sales.flowers.com"      "John Smith"
To:     WHORFIN
CC:
Subj:   Re: This is a test message.

Return-Path: <system@karem.yours.com>
Received: from karem.paul.com (192.168.1.92) by dino.bedrock.com
          (MX V5.1-X A2w8g) with SMTP for <smith@paul.yours.com>;
          Mon, 9 Aug 2014 14:35:01 -0400
Received: by karem.paul.com for smith@water.peter.com;
          Mon, 9 Aug 2014 14:35:00 GMT
Date: Mon, 9 Aug 2014 14:35:00 GMT
From: system@karem.paul.com
To: smith@water.peter.com
Message-ID: <990809143500.a2@karem.paul.com>
Glad to see your test worked.
This is my response.

MAIL>EXIT
```

Specifying a Host Alias

TCPware allows a system to have multiple names-or *host aliases*-with respect to electronic mail delivery. You can specify the host alias you want to use by defining the TCPWARE_SMTP_FROM_HOST logical name.

The alias you choose must be one of the SMTP host name aliases registered on the system (see the translation of the logical name TCPWARE_SMTP_HOST_NAME and the contents of the file TCPWARE_HOST_ALIAS_FILE). If the alias you use is unknown, the setting of TCPWARE_SMTP_FROM_HOST is ignored.

The host alias feature allows users from different administrative units within an organization to have their return address reflect the name of their unit, even though mail for all units is handled by one system.

Specifying Individual Aliases

TCPware supports both *system-wide* and *per-user* mail aliases. Using these aliases, you can refer to electronic mail addresses with names that are meaningful to you. Per-user mail aliases are kept in the file SMTP_ALIASES. in your login directory.

The format for alias entries is:

```
alias:     real_address[,...];
```

where *alias* is an alphanumeric string and *real_address* is an electronic mail address. You can specify multiple addresses by separating them with commas (,). The alias definition may span multiple lines, if needed, and must always be terminated with a semicolon (;).

For example, a local user may have a user name of JB134A, but you want to send mail to him as john. Add the following line to your SMTP_ALIASES. file:

```
john:      jb134A;
```

Aliases are repeatedly translated until no more translations are found. You can circumvent the repeated translations by including a leading underscore (_) in the *real_address*. For example, this definition causes mail to be forwarded and delivered locally:

```
fnord:     fnord@somewhere.else.edu, _fnord;
```

Using Mail under ALL-IN-1

This section explains how to use the mail subsystem under ALL-IN-1 to send mail to and receive mail from users on remote systems.

To send mail to a user on a remote system, specify an ALL-IN-1 e-mail address in the format:

```
recipient@destination@SMTP
```

@SMTP indicates to the ALL-IN-1 mail subsystem that the message should be given to the SMTP/MR gateway facility for eventual handling by the TCPware SMTP mail system. Note that the string SMTP and the destination system name are not case-sensitive; that is, you can type them in either uppercase or lowercase letters. However, the destination recipient specification may be case-sensitive, depending on the destination system's software. On some UNIX systems, ROOT and root specify two different user names (and hence different electronic mail addresses).

You receive network mail as you would all other mail in the ALL-IN-1 mail subsystem. Contact your system manager for the correct syntax for remote users; frequently, the proper syntax is:

```
yourname@A1.yourdomain
```

Delivering Mail to Specific Folders

The SMTP server supports incoming mail delivery to folders other than the NEWMAIL folder. The foldernames are restricted to UPPERCASE characters only, the pound sign (#), and the underscore (_). Use of the comma (,) in a foldername causes an error. Mail addressed to *user+folder@host* is delivered to the specified *folder*.

Note! Your system manager can disable this feature by defining the system-wide logical name TCPWARE_SMTP_DISABLE_FOLDER_DELIVERY.

User-Defined Headers

You can further customize your messages by defining special RFC 822 message headers.

SMTP-OpenVMS supports defining certain message header fields in the RFC 822 part of the message header.

Defining RFC 822 headers involves running the TCPWARE:CONFIG_SMTP_HEADERS.COM command file to define the following headers:

- Full name
- Comments
- Reply-to
- Return-receipt-to
- Bcc
- Sender
- X-Department
- X-Special user-defined header

Run the command procedure:

```
$ @TCPWARE:CONFIG_SMTP_HEADERS
```

The procedure checks for the TCPWARE_SMTP_USER_HEADERS logical for header definitions. If it does not find the logical, it checks the SYS$LOGIN:SMTP_USER_HEADERS.COM file. If it finds the file, it comes back with the prompt:

```
SYS$LOGIN:SMTP_USER_HEADERS.COM Exists.  Load? [Yes]
```

If you want to accept the contents of the file, press **Return**. (If the file did not load properly, you can have it overwritten at the next prompt.) You then have the choice of adding to, modifying, or deleting the file, exiting and saving, or quitting without saving:

```
[A]dd, [M]odify, [D]elete, e[X]it and Save or [Q]uit:
```

- If you are adding a header, the following prompt appears:

```
Add Header:

1. Full-Name:
2. Comments:
3. Reply-To:
4. Return-Receipt-To:
5. Bcc:
6. Sender:
7. X-Department:

8. Other

Which header would you like to add?
```

Enter the negative number value:

-1—Enter your full name
-2—Enter a comments line
-3—Enter a reply-to name address

-**4**—Enter a return-receipt-to name or address

A return-receipt-to value is only valid if the system logical TCPWARE_SMTP_RETURN_RECEIPT_TO_HEADER_ENABLE is defined as `1` during configuration. If this system logical is not defined or defined as `0`, SMTP-OpenVMS does not add the `Return_receipt_to` header to the mail message.

-**5**—Enter a `Bcc:` name or address
-**6**—Enter a sender name or address
-**7**—Enter a departmental name or address

SMTP-OpenVMS prepends an `X-` to the departmental name or address.

-**8**—Enter your own special header. For example:

```
What is the name of the header: X-Affiliation
```

SMTP-OpenVMS prepends an `X-` to the special header name.

The next prompt asks you to supply a value for the header you specify. For example:

```
Full-Name Value: George Plimpton
```

The procedure returns to the `[A]dd, [M]odify, [D]elete, e[X]it and Save or [Q]uit:` prompt so that you can add other headers or modify or delete existing ones. If you enter **`X`** (exit and save), the procedure writes out the file on exiting and defines the SMTP_USER_HEADERS logical based on the file's contents.

- If you are modifying a header definition, the procedure gives you the current list of defined headers, followed by a prompt, where you enter the appropriate number. For example:

```
Your Current Headers:

1. Full-Name Value: George Plimpton
2. X-Affiliation: Paris Review

Which header would you like to modify? 2

New X-Affiliation Value: None
```

After modification, you return to the `Which header would you like to modify?` prompt. If you enter **`Return`** at the prompt, you return to the `[A]dd,`... prompt.

- If you are deleting a header definition, the procedure gives you the current list of defined headers, followed by the prompt:

```
Which header would you like to remove?
```

The procedure asks for confirmation and returns to the above prompt unless you enter **`Return`**. Removed files show up as being deleted in the `Your Current Headers:` list until you add a new header, or exit and reenter the procedure.

Chapter 11 TALK: Exchanging Terminal Messages

Introduction

The TALK utility allows you to exchange messages you type at your terminal with another local or remote user. You do not need to wait between sending your message and receiving one from your destination user. TALK uses a split screen where what you type is on the top half and what the other person types is on the bottom. This allows you to talk in "real time."

Using TALK

First make sure the OpenVMS Phone Utility is on. If you show the broadcast status for your terminal and get something like the following:

```
$ SHOW BROADCAST
Broadcasts are currently disabled for:
    PHONE
    MAIL
    QUEUE
    SHUTDOWN
Then you enable phone broadcasting as follows:
```

```
$ SET BROADCAST=PHONE
```

To set up and invoke TALK, enter at the DCL prompt:

```
$ TALK:==$TCPWARE:TALK.EXE
$ TALK username[@host] [ttyname]
If you are communicating with another local user, type the user's username. If
communicating with a user on another system, use the username@host syntax.
```

You can also include the terminal port (*ttyname*) as a parameter. Most UNIX servers only ring one of and not all the remote user's terminals. If the remote user is logged in many times, and you would rather ring a terminal that has been idle for only a short period, specify the terminal port using *ttyname*.

One way to discover terminal ports is by using the FINGER utility, such as in the following example, where there are two terminal ports, `ttyp5` and `ttyp7`. Since the `ttyp7` terminal has a much shorter idle time (and is more current), it is therefore a better candidate for a TALK terminal:

```
$ FINGER MARGE@MARGE.ZOZO.COM
Login name: marge                       In real life: Marge Simpson
Directory: /home/spectre                Shell: /usr/local/bin/tcsh
On since Nov 3 10:06:48 on ttyp5 from bart.nene.com
59 minutes Idle Time
```

```
Login name: marge                       In real life: Marge Simpson
Directory: /home/spectre                Shell: /usr/local/bin/tcsh
On since Nov 3 10:06:44 on ttyp7 from bart.nene.com
36 seconds Idle Time
```

```
$ TALK MARGE@MARGE.ZOZO.COM TTYP7
```

After the above command, TALK sends the following message to the recipient if the connection is successful:

```
Message from Talk_Daemon@destination-host...
talk: connection requested by yourname@yourhost.
talk: respond with: talk yourname@yourhost
```

To establish the connection, the recipient follows the instructions from the Talk_Daemon and types the following at the system prompt:

```
talk yourname@yourhost
```

It does not matter from which machine the recipient replies, as long as the recipient's login name is the same. Once communication is established, the two parties can type simultaneously, with their output appearing in two parts of a split screen. What you type appears on the top half and what the other person types is on the bottom half of the screen.

To signal that you are expecting a response, it is customary to leave a blank line after your last line of text. You can use a convention such as **-oo** ("over and out") to signal that your part of the correspondence is over.

Type **Ctrl/L** to reprint the screen. You can also use the erase, kill, and word kill (**Ctrl/K**) characters.

To exit, type the interrupt character (**Ctrl/C**, **Ctrl/Y**, or **Ctrl/Z**). TALK moves the cursor to the bottom of the screen and restores the terminal to its previous state.

Example 11-1 shows a sample exchange between user BART on host BART.ZOZO.COM (an OpenVMS system) and user MARGE on host MARGE.NENE.COM.

Example 11-1 Sample TALK Message Exchange

On Bart:

(Bart) $ **TALK:==$TCPWARE:TALK.EXE**

(Bart) $ **TALK MARGE@MARGE.NENE.COM**

------------[Waiting for your party to respond]---------------------

On Marge:

(Marge) $

```
Message from Talk_Daemon@BART.ZOZO.COM at 11:23 ...
talk: connection requested by bart@bart.zozo.com.
talk: respond with:  talk bart@bart.ZOZO.com
(Marge) $ TALK BART@BART.ZOZO.COM

------------[Connection established: bart@bart.zozo.com]-----------
```

On Bart:

```
Hi, there!

------------[Connection established:
```

```
marge@marge.nene.com]-----------
```

On Marge:

```
Good to hear from you!

------------[Connection established: bart@bart.zozo.com]--

Hi, there!
```

On Bart:

```
Hi, there!

See you soon! -oo   Ctrl/C

------------[Connection closed.
Exiting]-----------------------------
Good to hear from you!

(Bart) $
```

Command Reference

The following is a command reference to the TALK utility.

TALK

The TALK command is a visual communication program that exchanges messages with another host user by copying lines you type on your terminal to the other user's terminal. The other host recipient must support the `ntalk` protocol to accept (and respond to) your messages.

It does not matter from which machine the recipient replies, as long as the recipient's login name is the same. Once communication is established, the two parties can type simultaneously, with their output appearing in different parts of the same window.

Typing **`Ctrl/L`** causes the screen to be reprinted, while the erase, kill, and word kill (**`Ctrl/K`**) characters work in TALK as normal.

To exit, type your interrupt character (**`Ctrl/C`**, **`Ctrl/Y`**, or **`Ctrl/Z`**). TALK moves the cursor to the bottom of the screen and restores the terminal.

Format

TALK *username[@host] [ttyname]*

Parameters

username[@host]

If you want to talk to someone on your own machine, *username* is just the local user's login name. If you want to talk to a user on another host, use the form *username@host*.

ttyname

Name of the specific remote terminal. Many UNIX clients do not send `talk` request messages to every terminal of the user, and usually select just one. You may, however, want to make a particular selection.

Restrictions

This version of TALK is incompatible with versions of ULTRIX earlier than v3.0. Starting with ULTRIX v3.0, TALK communicates with other machines running ULTRIX v3.0 (and later), and machines running 4.3BSD or versions of UNIX based on 4.3BSD.

TALK is not eight-bit clean. Typing in DEC Multinational Characters (ISO-8859/1) causes the characters to echo as a sequence of carets (^) followed by the character represented with its high bit cleared. This limitation makes TALK unusable if you want to communicate using a language that has ISO-8859/1 characters in its alphabet.

Example

```
system1>talk user2@system2
```

The following message appears on the screen of `user2`:

```
Message from Talk_Daemon@system2 at 12:37 ...
talk: connection requested by user1@system1.
talk: respond with: talk user1@system1
```

To establish the connection, `user2` follows the instructions from the `Talk_Daemon` and types the following at the system prompt:

```
system2>talk user1@system1
```

Troubleshooting

The **`Your party is refusing messages`** message may come up if the remote terminal is set up with messages off, such as the first terminal (`tty5`) in the following example:

```
(Bart) FINGER MARGE@MARGE.ZOZO.COM
Login name: marge     (messages off)    In real life: Marge Simpson
Directory: /home/spectre                Shell: /usr/local/bin/tcsh
On since Nov 3 10:06:48 on ttyp5 from bart.nene.com
59 minutes Idle Time
No unread mail
No Plan.

Login name: marge                       In real life: Marge Simpson
Directory: /home/spectre                Shell: /usr/local/bin/tcsh
On since Nov 3 10:06:44 on ttyp7 from bart.nene.com
36 seconds Idle Time

(Bart) TALK MARGE@MARGE.ZOZO.COM TTYP5

----------------[Your party is refusing messages]--------------------
```

The **`Checking for invitation on caller's machine`** message may come up when the client is waiting for a response from the remote system. If the message appears for an extended time, it may mean that the remote system's server does not support the `ntalk` protocol, in which case a connection is not possible.

If the message **`Your party is not logged on`** appears, the remote user is not logged on at the time.

Chapter 12
TELNET: Connecting to Remote Terminals

Introduction

The Virtual Terminal Protocol (TELNET) provides connections to remote hosts. With it, you can access remote hosts using OpenVMS commands or a UNIX style command interface.

The Client-TELNET utility is your interface to TELNET OpenVMS. You can run Client-TELNET interactively or through a startup command procedure.

Client-TELNET supports normal and TN3270 mode:

- Normal mode uses your local OpenVMS keyboard. In this mode, you can open up to ten TELNET sessions at one time.
- In TN3270 mode, Client-TELNET emulates the keyboard normally used on an IBM 3270-class terminal. It allows you to connect only one TN3270 session at a time.

Before Using TELNET

Before you use TELNET, ask your system manager if the TELNET-OpenVMS software was installed, configured, and started on your system.

To use TELNET with Kerberos version 4 authentication, your system manager must have configured TCPware's Kerberos Services.

Before you can connect to a remote host, you need to know:

- The name of the remote host to which you want to connect.
- The username and password for each account on the remote host. If the remote host does not support multiuser protection features, you may not need a username and password. If you are using TCPware's Token Authentication, you also need to enter the additional PASSCODE from your SecurID$^{©}$ token (see theOPEN command for details).
- How to use the operating system of the remote host.

Note! Client-TELNET does not restrict the ASCII character set to seven-bit ASCII as the TELNET standard implies. Client-TELNET supports the full eight-bit (multinational) character set. To use the multinational character set, you must configure your terminal to support eight-bit characters. The peer TELNET implementation must also support the same.

Opening a TELNET Session

Run the Client-TELNET utility to connect to a remote host. Client-TELNET supports as many as 10 connected sessions at any one time. However, of these ten sessions, only one can be a TN3270 session. To open a TELNET session (see Example 12-1):

1 At the DCL prompt, enter: `$ TELNET`

2 Use the OPEN command to open a remote TELNET session in one of the following ways:

a To use standard authentication, at the TELNET> prompt, enter either:

```
TELNET>OPEN host
TELNET>OPEN host /AUTH=NULL
```

—*host* is the name of the host to which you want to connect. /AUTH=NULL explicitly specifies to use standard authentication.

b To use Kerberos version 4 authentication, enter at the `TELNET>` prompt:

TELNET> **OPEN *host* /AUTH=KERBV4 /REALM=*realm***

—*host* is the name of the host to which you want to connect.

–`/AUTH=KERBV4` specifies the use of Kerberos version 4 authentication.

–`/REALM=realm` specifies the name of the Kerberos Server realm.

You must first get a ticket-granting ticket (TGT) from the Kerberos Server. (See Chapter 4, *Kerberos User Commands*.)

You can specify the Kerberos realm using the /REALM qualifier. If you omit the qualifier, the contents of the TCPWARE:KRB.REALMS file determines the Kerberos realm.

To open a connection, TELNET first tries to use Kerberos version 4 authentication if requested, then reverts to standard authentication if Kerberos version 4 authentication fails.

3 Respond to the login prompts, if any, of the remote host, including any PASSCODE.

4 Open another session if desired:

a Return to the local TELNET prompt by entering the escape sequence displayed when opening the connection (usually **`Ctrl/\`**). The previous session remains open.

b Use the OPEN command to open the next session. Repeat steps 2 and 3.

Alternative method. You can also open a remote TELNET connection as follows:

```
$ TELNET host
```

See the OPEN, CLOSE, and EXIT commands in the *Command Reference*.

Example 12-1 Opening Multiple TELNET Sessions

```
(IRIS) $ TELNET
TELNET>OPEN BART
%TCPWARE_TELNET-I-TRYING, trying bart.nene.com,telnet(192.168.1.92,23)...
%TCPWARE_TELNET-I-ESCAPE, escape (attention) character is "^\"
```

(login procedure to BART)

```
(BART) $ Ctrl/\

TELNET> OPEN MARGE                   [BART remains open]
%TCPWARE_TELNET-I-TRYING, trying marge.nene.com,telnet
(192.168.1.91,23)...
%TCPWARE_TELNET-I-ESCAPE, escape character is "^\"
```

(login procedure to MARGE)

```
(MARGE) $ Ctrl/\

TELNET>OPEN HOMER                                 [BART and MARGE remain open]
%TCPWARE_TELNET-I-TRYING, trying homer.nene.com,telnet
(192.168.1.90,23)...
%TCPWARE_TELNET-I-ESCAPE, escape character is "^\"
```

(login procedure to HOMER)

```
(HOMER) $ Ctrl/\

TELNET> OPEN LISA       [BART, MARGE, and HOMER remain open]
%TCPWARE_TELNET-I-TRYING, trying lisa.nene.com,telnet
(192.168.1.89,23)...
%TCPWARE_TELNET-I-ESCAPE, escape character is "^\"
```

(login procedure to LISA)

```
(LISA) $ Ctrl/\

TELNET> OPEN /AUTH=KERBV4 /REALM=SIMPSONS.COM MAGGIE
%TELNET-I-TRYING, trying maggie.yours.com,telnet (192.168.99.1,23)...
%TELNET-I-ESCCHR, escape (attention) character is "^\"
(MAGGIE) $
```

Note! TCPware provides secure TELNET-OpenVMS logins through its Token Authentication feature, if installed and enabled. For more information, see Chapter 14, *Token Authentication: Protecting Logins*.

Opening a TN3270 Session

Client-TELNET supports TN3270 mode for local OpenVMS terminals. The remote IBM host must support a TELNET server.

You can only connect one TN3270 session at any one time. Client-TELNET returns an error message if you try to open more than one TN3270 session.

To open a TELNET session in TN3270 mode (see Example 12-2):

1 At the DCL prompt, enter: `$ TELNET`

2 Use the OPEN command at the TELNET> prompt: `TELNET>OPEN host [/TN3270]`

TELNET servers that cannot automatically negotiate this mode require the /TN3270 qualifier.

3 Enter the TN3270 escape sequence `Ctrl/C` instead of `Ctrl/\`.

4 If you want to print a screen in TN3270 mode, add the /PRINT qualifier as follows:

TELNET>**OPEN *host* /TN3270 /PRINT=(FILE=*filename* | QUEUE=*qname*)**

SeeTN3270 Screen Printing and Dumping.

5 Only one TN3270 session can be open at any given time. If you try to open more than one TN3270 session, Client-TELNET returns an error message.

Table 12-1 lists the IBM terminal models and screen sizes Client-TELNET supports. To use the emulated model, your terminal must support the minimum size (number of rows and columns) indicated. DECwindows, DECterm, and virtual workstation (VWS) windows resize accordingly.

Table 12-1 Supported IBM Models

Emulated Model	Minimum Size (rows x columns)
IBM 3278-2	24 x 80
IBM 3278-3	32 x 80
IBM 3278-4	43 x 80
IBM 3278-5	27 x 132

Some Client-TELNET commands have specific meaning for TN3270 mode.

See *TN3270 Keyboard Mapping*.

Alternative method. You can also open a remote TELNET TN3270 connection by entering the following command:

```
$ TELNET host /TN3270
```

See the OPEN, CLOSE, and EXIT commands in the *Command Reference*.

Example 12-2 Opening a TN3270 Session

```
$ TELNET
TELNET>OPEN LOCIS.LOC.GOV
<Library of Congress menus displayed>
Ctrl/C

TELNET>CLOSE
TELNET>OPEN LOCIS.LOC.GOV /TN3270 /PRINT(=QUEUE=ENG_PRINTER_ASCII)
Ctrl/C

TELNET>OPEN BLUE.ADP.WISC.EDU /TN3270
%TCPWARE-TELNET-E-CONLOST, connection to remote host lost
%TCPWARE-TELNET-E-MAXTN3270, only one TN3270 session may be open at any
one time
```

```
%TCPWARE-TELNET-I-CURRSESSION, current session is not 1, LOCIS.LOC.GOV
TELNET>
```

Closing a Session

A TELNET session remains open until you log out of that session at the system prompt or use the CLOSE, EXIT, QUIT, or BYE commands or enter **`Ctrl/Z`** at the `TELNET>` prompt.

To close a TELNET session, use one of the following commands at the `TELNET>` prompt (see Example 12-3):

- `TELNET>`**`CLOSE`** closes the current session, as in the following chart:

If you open a TELNET session using...	And...	Then CLOSE closes the current session and...
`Telnet>`**`OPEN`** ***`host`***	It is the only session There are other sessions	Keeps you in TELNET Keeps you in TELNET with the other sessions open
`$` **`TELNET`** ***`host`***	It is the only session There are other sessions	Exits TELNET Keeps you in TELNET with the other sessions open

If you close the current session, and there are other connected sessions, Client-TELNET resets the current session to the "next" session.

- `TELNET>`**`CLOSE`** ***`session-number`***
  closes only the specified session, as indicated by the SHOW STATUS command.
- `TELNET>` **`EXIT`** exits TELNET
- `TELNET>`**`QUIT`** exits TELNET
- `TELNET>`**`BYE`** exits TELNET
- `TELNET>`**`Ctrl/Z`** interrupts TELNET

See the OPEN, CLOSE, EXIT, and SHOW STATUS commands in the *Command Reference*.

Example 12-3 Closing TELNET Sessions

```
(IRIS) $ TELNET
TELNET>OPEN BART
%TCPWARE_TELNET-I-TRYING, trying bart.nene.com,telnet(192.168.1.92,23)...
%TCPWARE_TELNET-I-ESCAPE, escape character is "^\"
```

(login procedure to BART)

```
(BART) $ Ctrl/\
TELNET> OPEN MARGE      [BART remains open]
%TCPWARE_TELNET-I-TRYING,trying marge.nene.com,telnet(192.168.1.91,23)...
%TCPWARE_TELNET-I-ESCAPE, escape character is "^\"
```

(login procedure to MARGE)

```
(MARGE) $ Ctrl/\
TELNET>SHOW STATUS
Client-TELNET V6.0-0 Copyright (c) Process Software
Connected sessions:
     1. bart.nene.com,telnet (192.168.1.92,23).
 --> 2.  marge.nene.com, telnet (192.168.1.91,23).
"^\" is the escape (attention) character

TELNET> CLOSE 2
%TCPWARE_TELNET-I-CONNCLOSED, closing session 2, marge.nene.com
TELNET>CLOSE 1
%TCPWARE_TELNET-S-CONNCLOSED, closing session 1,bart.nene.com
TELNET>EXIT
(IRIS) $
```

Issuing Local Commands

You can issue commands to the Client-TELNET utility during a remote session by returning to the TELNET prompt. You can then enter one or more TELNET commands.

TELNET OpenVMS features multiline recall of up to 20 command lines using the standard OpenVMS line recall and editing keys.

You return to the remote session by entering the RESUME command.

To issue a local TELNET command while connected to a remote host and then resume the session on the host (see Example 12-4):

1 Enter the escape (attention) character to return to the TELNET prompt: for example: **`Ctrl/\`**

2 Issue a TELNET command. For example, you may want to:

- Issue the SHOW STATUS command. The SHOW STATUS command displays a list of open connections. The arrow (-->) identifies the current session.

Change the escape (attention) character using the SET ESCAPE command.

3 Return to the remote host by entering: `TELNET>`**`RESUME`**

This command resumes to the current remote host. Pressing **`Return`** or entering the OPEN command also resumes to the current remote host.

To resume to a different session, enter: `TELNET>`**`RESUME`** ***`session-number`***

– *session-number* is the number of the session which you want to resume. The session-number refers to a particular connection, as displayed by the SHOW STATUS command.

You can switch between local TELNET command mode and the remote host as often as you like.

See the RESUME, SET ESCAPE, and SHOW STATUS commands in the *Command Reference*.

Example 12-4 Issuing TELNET Commands and Resuming a Session

```
(BART) $ Ctrl/\

TELNET>SHOW STATUS
Client-TELNET V6.0-0 Copyright (c) Process Software
Connected sessions:
          1. BART.nene.com,  telnet  (192.168.1.92,23).
                 2. HOMER.nene.com, telnet  (192.168.1.90,23).
                 3. MARGE.nene.com, telnet  (192.168.1.91,23).
          -->    4. LISA.nene.com,  telnet  (192.168.1.89,23).
```

```
"^\" is the escape (attention) character.

TELNET>SET ESCAPE "^A"
escape (attention) character is "^A"

TELNET>RESUME
(BART) $

(BART) $ Ctrl/\

TELNET>RESUME 2
%TCPWARE_TELNET-I-RESUME, resuming session 2, HOMER.illiad.com
(HOMER) $
```

Running Applications over TELNET

You can run applications over a TELNET connection by creating an NTA terminal on the local client. You can only create such devices from TELNET with no other escaped connection. This section describes how to create non-permanent NTA devices. To create permanent NTA devices, see the next section.

Normally, Client-TELNET connects to an NTA device at the TCPware server end of the connection. It does not usually create a local NTA device. However, you can create a local NTA device so that you can run applications over the TELNET connection. To create a local NTA device (see Example 12-5):

1 Enter at the DCL prompt one of the following:

- `$` **TELNET *host* /CREATE**
- `TELNET>`**OPEN /CREATE**

Use the second method if you already logged in to a host, and escaped from the session (using **Ctrl/** or some other defined escape sequence).

In both cases, this associates a pre-allocated local NTA*x*: terminal device to your TELNET connection; *x* is the next available unit number. No other escaped connection can exist during your TELNET session for this to work. (If one exists, the `%TCPWARE_TELNET-E-CONNOPN` error message appears.)

2 Run your application at the DCL prompt. Use the allocated terminal device as desired.

3 When your application ends, clean up by deallocating the NTA device you created using the following command at the DCL prompt: `$` **DEALLOCATE *device***

See your OpenVMS documentation for details on the DEALLOCATE command.

Note! Using /CREATE in this way creates a non-permanent NTA device, which has certain ramifications. See the next section for details on how to create a permanent NTA device. Using the OPEN /CREATE command as part of a TELNET command file creates an NTA device and exits TELNET right away without passing any further commands in the file to TELNET.

You can also invoke TELNET and use OPEN/CREATE non-interactively, such as with a batch file. The batch file cannot open an interactive connection. For applications run by the creating process, use the /LOGICAL qualifier to create a predefined name for the device. If this device is to be used by another process, the qualifier /LOGICAL=... /TABLE= may help reference it. For example:

```
$ TELNET SIGMA /CREATE /LOGICAL=TELNET_NTA /TABLE=SYSTEM /MODE=EXEC
```

See the OPEN command in the *Command Reference* for other parameters you can use with the /CREATE qualifier.

Example 12-5 Opening a TELNET Connection to a Terminal Device

```
$ TELNET MARGE /CREATE
%TCPWARE_TELNET-I-TRYING, trying marge.nene.com,telnet
(192.168.1.91,23)...
%TCPWARE_TELNET-I-ALOC,_MARGE$NTA1: allocated

$ SET HOST/DTE NTA1:
$ DEALLOCATE NTA1:


$ TELNET BART
%REM-I-TOQUIT, connection established

Press Ctrl/\ to quit, Ctrl/@ for command mode

OpenVMS VAX 5.-2 with TCPware for OpenVMS 5.6

Username: Ctrl/\
TELNET>OPEN /CREATE
%TCPWARE_TELNET-I-ALLOC, _NTA1: allocated
$ SET HOST/DTE NTA1:
$ DEALLOCATE NTA1:
```

Creating a Permanent NTA Device

You can also run applications over a TELNET connection by creating a permanent NTA terminal on the local client. This permanent device acts more like a LAT device; it is not automatically deleted when there are no process channels assigned to it, it can be handed off to other applications, and it has reconnect capabilities in case of a connection break. This section describes how to create permanent NTA devices. To create non-permanent NTA devices, see the previous section.

Using TELNET /CREATE by itself to create a non-permanent NTA device, such as in the previous section, has the following limitations:

- An application using this NTA device may be written to deassign and thus delete the device if the connection goes down. This could cause a conflict when rerunning the application if, meanwhile, another NTA connection with the same unit number is created.
- Handing off the NTA device to another process may require setting up the device as NOHANGUP.
- Recovery is not possible in case of a broken connection.

You can bypass these limitations and make the NTA device a permanent one by adding the PERMANENT keyword to the TELNET /CREATE command, as follows (see Example 12-6):

Note! Creating a permanent NTA device requires OPER privilege.

1 Enter at the DCL prompt:

```
$ TELNET host port /CREATE=PERMANENT
```

or:

```
TELNET>OPEN host port /CREATE=PERMANENT
```

This creates a permanent local NTAx: terminal device with the next available unit number. However, unlike non-permanent NTA devices, the TELNET utility does not pre-allocate it. Likewise, you can specify the /LOGICAL qualifier to set up a logical name for the device so that other applications can use it.

It is advisable that you specify a port other than the default TELNET port 23.

See the OPEN command in the *Command Reference* section for other parameters you can use with the /CREATE=PERMANENT qualifier.

2. Run your application at the DCL prompt, as with a non-permanent NTA device. The difference is that handing off the NTA device to another process and recovery of a broken connection are enhanced.
3. In handing off the NTA device to another process, you may wish to change its protection:
   - In VMS 5, use SET PROTECTION, or SET DEVICE /ACL.
   - VMS 6 replaces these commands with SET SECURITY /PROTECTION= /ACL.

Example 12-6   Setting up a Permanent NTA Device

```
TELNET>OPEN MARGE 7 /LOGICAL=MY_PORT -
_TELNET>/CREATE=(PERMANENT,INTERVAL=10,RETRIES=10)
%TCPWARE_TELNET-I-CREATED, _NTA1: created

$ @MY_APPLICATION MY_PORT
```

Note! For information on /LOGICAL= *qualifier*, see 12-33.

Handling a Broken Connection

If the connection to the remote port is broken, a temporary NTA device is reported as "Offline" with $QIO's failing with a SS$_DEVOFFLINE status. For a permanent NTA device, however:

- The NTA devchar is marked UNAVAILABLE (which can be viewed by using SYS$GETDVI to check if DVI$_DEVCHAR's DEV$V_AVL = 0).
- If a terminal Ctrl/Y AST is set up, the AST fires up. (Setup: Disable Ctrl/Y handling by DCL using LIB$DISABLE_CTRL(&LIB$M_CLI_CTRLY, 0), and set up the AST using SYS$QIOW with IO$_SETMODE | IO$M_CTRLYAST.)
- Terminal I/Os queued in the TTdriver are completed with the I/O Status Block (IOSB) having a status of SS$_HANGUP.
- A new write $QIO buffers the data so that it can be sent when reconnected. If no reconnection is being done, then one is set up.
- Data sent at the time of the broken connection may be lost.
- The client attempts to reconnect to the remote port as described in the OPEN /CREATE=PERMANENT command section.
- The permanent NTA handles reconnects internally instead of allowing the program to issue the LAT SYS$QIOW with IO$TTY_PORT | IO$M_LT_CONNECT.

Closing the Connection After a Deassign

You can use the CLOSE_DASSGN keyword to the /CREATE=(PERMANENT) qualifier to close the underlying TCP connection when the last channel assigned to the NTA device is dropped using SYS$DASSGN. The default is not to close the TCP connection.

Startup Command File

You can have a startup file executed each time you invoke Client-TELNET. The TELNET_STARTUP logical specifies a file that contains commands you want performed at the beginning of each TELNET session.

To set up and run a startup command file (see Example 12-7):

1. Create a TELNET_STARTUP.COM file in your login directory.
2. In the file, include the TELNET command or commands you want executed each time you start Client-TELNET.

3 Edit your LOGIN.COM file and define the TELNET_STARTUP logical name to point to the startup file. For example, add the following line to your login file:

```
$ DEFINE/PROCESS TELNET_STARTUP "SYS$LOGIN:TELNET_STARTUP.COM"
```

4 Rerun LOGIN and run TELNET.

Whenever you run TELNET OpenVMS, it first looks for the file to which the TELNET_STARTUP logical points. It then processes all the commands contained in that file until it processes the EXIT command or reaches the end of the file.

If the OPEN command appears in this file, TELNET establishes the connection and all further input comes from the terminal. When you return to command mode, TELNET processes the rest of the commands in the startup file (if any).

If the EXIT command appears in the startup file, Client-TELNET ignores all commands following the EXIT command and continues TELNET operations, leaving the user at the TELNET prompt.

Example 12-7 Setting Up a Startup Command File

```
$ CREATE TELNET_STARTUP.COM
SET TRANSLATION /SEND=CR
OPEN IRIS
OPEN HOMER
SHOW STATUS
Ctrl/Z

$ EDIT SYS$LOGIN:LOGIN.COM
$ DEFINE/PROCESS TELNET_STARTUP "SYS$LOGIN:TELNET_STARTUP.COM"
Ctrl/Z

$ @SYS$LOGIN:LOGIN
$ TELNET
TELNET>SET TRANSLATION /SEND=CR
%TCPWARE_TELNET-I-TRNSNEWLN, will translate CR to CRLF when sent
TELNET>OPEN IRIS
%TCPWARE_TELNET-I-TRYING, trying IRIS.plants.com,telnet
 (192.168.1.93,23) ...
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^\"

(login procedure to IRIS)

(IRIS)$ Ctrl/\
TELNET>OPEN HOMER
%TCPWARE_TELNET-I-TRYING, trying HOMER.illiad.com,telnet
(192.168.1.90,23) ...
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^\"

(login procedure to HOMER)

(HOMER)$Ctrl/\
TELNET>SHOW STATUS

Connected sessions:
1. IRIS.plants.com, telnet         (192.168.1.93,23).
-->2. HOMER.illiad.com, telnet     (192.168.1.90,23).
"^\" is the escape (attention) character.

No characters are translated to CRLF when received.
CR is translated to CRLF when sent.
```

```
TELNET>RESUME
(HOMER)$
```

TN3270 Keyboard Mapping

When the current Client-TELNET session is in TN3270 mode, Client-TELNET lets your local OpenVMS keyboard emulate the keyboard normally used on an IBM 3270-class terminal. The TCPWARE:MAP3270.DAT file defines the key mappings. The MAP3270.DAT file supports all the standard HP terminal types.

If you have a non-standard terminal, make sure the TCPWARE:MAP3270.DAT file and the OpenVMS SYS$SYSTEM:TERMTABLE.TXT file contain the appropriate keyboard definitions. If you need to alter definitions in the MAP3270.DAT file, note the following:

- MAP3270.DAT is not case-sensitive. IBM-to-OpenVMS Keyboard Map lists the key mapping in this file.
- One entry contains all key definitions for a particular terminal.
- Use this format to define each key:

***key-name* = '*key-sequence*' ['*key-sequence*'];**

– *key-name* is a key name defined in the MAP3270.DAT file.

– *key-sequence* is the sequence of OpenVMS keys used to perform the IBM function.

- Use the following conventions when you alter key map definitions:

Convention	**Meaning**
{ }	Encloses each entry
' '	Encloses key sequences. For example: `'^m'`
|	"or." For example: `'^z'` | `'\EOM'`
^	Introduces a control character. For example: `'^z'`
\n	Newline
\t	Tab
\r	Carriage return
/E	Escape
\'	Represents a single quote when used in a key definition
;	Ends a key definition. For example: `'^z'` | `'\EOM';`

#	Begins a comment

SeeTN3270 Keypad Graphics Characters.

Alternative key mappings

Client-TELNET provides an alternative mapping file that closely resembles the keyboard mappings provided by the OpenVMS DECwindows DECnet/SNA 3270 Terminal Emulator. To use these mappings, redefine the TCPWARE_TELNET_KEYBOARD_MAP logical to point to the MAP3270_DECSNA.DAT file. By default, this logical points to the MAP3270.DAT file.

You can also define your own key mapping file. Just make sure you redefine the TCPWARE_TELNET_KEYBOARD_MAP logical so that it points to the new file.

Table 12-2 IBM-to-OpenVMS Keyboard Map

<table>
<tr><th>IBM Function</th><th>OpenVMS Keys</th><th>IBM Function</th><th>OpenVMS Keys</th></tr>
<tr><td>Enter</td><td>Ctrl/M or <CR></td><td>PF15</td><td>Ctrl/F-1-5 or PF1-KP5</td></tr>
<tr><td>Clear</td><td>Ctrl/Z or Enter</td><td>PF16</td><td>Ctrl/F-1-6 or PF1-KP6</td></tr>
<tr><td>Newline</td><td>Ctrl/N</td><td>PF17</td><td>Ctrl/F-1-7 or PF1-KP7</td></tr>
<tr><td>Tab</td><td>Ctrl/I or Tab</td><td>PF18</td><td>Ctrl/F-1-8 or PF1-KP8</td></tr>
<tr><td>Backtab</td><td>Ctrl/B</td><td>PF19</td><td>Ctrl/F-1-9 or PF1-KP9</td></tr>
<tr><td>Left arrow</td><td>Ctrl/H or left arrow</td><td>PF20</td><td>Ctrl/F-2-0 or PF2-KP0</td></tr>
<tr><td>Right arrow</td><td>Ctrl/L or right arrow</td><td>PF21</td><td>Ctrl/F-2-1 or PF2-KP1</td></tr>
<tr><td>Up arrow</td><td>Ctrl/K or up arrow</td><td>PF22</td><td>Ctrl/F-2-2 or PF2-KP2</td></tr>
<tr><td>Down arrow</td><td>Ctrl/J or down arrow</td><td>PF23</td><td>Ctrl/F-2-3 or PF2-KP3</td></tr>
<tr><td>Home</td><td>KP. (keypad period)</td><td>PF24</td><td>Ctrl/F-2-4 or PF2-KP4</td></tr>
<tr><td>Delete</td><td>DEL or Remove</td><td>PA1</td><td>Ctrl/P-1 | ESC/PF1 | PF4</td></tr>
<tr><td>Erase to EOF</td><td>Ctrl/E</td><td>PA2</td><td>Ctrl/P-2 | ESC/PF2 | KP-
(keypad dash)</td></tr>
</table>

Erase input	Ctrl/W	PA3	Ctrl/P-3 | ESC/PF3 | KP, (keypad comma)
Insert	Ctrl-space, ESC-space or Insert Here	Escape to TELNET command	Ctrl/C
PF1	ESC/1 or KP1	Master reset	Ctrl/G
PF2	ESC/2 or KP2	Set tab	ESC/;
PF3	ESC/3 or KP3	Delete tab	ESC/’
PF4	ESC/4 or KP4	Clear tabs	ESC/:
PF5	ESC/5 or KP5	Set margin	ESC/,
PF6	ESC/6 or KP6	Set home	ESC/.
PF7	ESC/7 or KP7	Column tab	ESC/↓
PF8	ESC/8 or KP8	Column back tab	ESC/
PF9	ESC/9 or KP9	Indent	ESC/→
PF10	ESC/0 or PF1-KP0	Unindent	ESC/¬
PF11	ESC/-or PF1-KP1	Indent	ESC/→
PF12	ESC/ = or PF1-KP2	Indent	ESC/→
PF13	Ctrl/F-1-3 or PF1-KP3	Indent	ESC/→
PF14	Ctrl/F-1-4 or PF1-KP4	Indent	ESC/→

TN3270 Internationalization

International character set support adds functionality to convert the Western European EBCDIC character set to the corresponding terminal character sets (multinational or national replacement).

Since current TCPware TN3270 does not support the structured field of the extended terminals, this support does not add the simultaneous multiple character set functionality the extended terminals provide.

The TELNET command line and OPEN command include two qualifiers to support TN3270 Internationalization:

/HOST_CHARACTER_SET=*host-character-set-name*
/TERMINAL_CHARACTER_SET=*terminal-character-set-name*

- /HOST_CHARACTER_SET lets you specify the national EBCDIC character set. Table 12-3 contains the supported character sets and their corresponding IBM code page numbers.
- /TERMINAL_CHARACTER_SET lets you specify the character set used on the terminal (OpenVMS system) side. Table 12-4 includes the supported Multinational and National Replacement character set values.

You can also use logicals to specify the host/terminal character set selection. The system manager may choose to set up a system logical to specify the default character set for his site. The logicals are:

TCPWARE_TN3270_HOST_CHARSET—Host character set
TCPWARE_TN3270_TERMINAL_CHARSET—Terminal character set

You can specify the same values as you do with the corresponding qualifiers. For example:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_TN3270_HOST_CHARSET CANADIAN
$ DEFINE/SYSTEM/EXEC TCPWARE_TN3270_TERMINAL_CHARSET LATIN1
```

The TELNET SHOW STATUS command displays the currently selected character set for TN3270. For example:

```
TELNET>SHOW STATUS
Client-TELNET V6.0-0  Copyright (c) Process Software
Connected session:

  -->1. LOCIS.LOC.GOV, telnet (140.147.254.3,23). [TN3270 mode]

Current session is operating in 3270 mode.
Terminal type:  IBM-3278-2

Keyboard Map File:  TCPWARE:MAP3270.DAT

Host Character Set: CANADIAN
Terminal Character Set: LATIN1

Print key function:
Output File: SYS$LOGIN:TN3270.TXT

"^C" is the escape (attention) character.
```

Table 12-3 TN3270 Internationalization Character Sets

Character Set	Code Page	Character Set	Code Page
AUSTRIAN	273	INTERNATIONAL	038
BELGIAN	274	NORWEGIAN	277
CANADIAN	037	PORTUGUESE	037

DANISH	277	SPANISH	284
DUTCH	037	SWEDISH	278
ENGLISH_UK	285	SWISS	500
ENGLISH_US	037	FRENCH	297
FINISH	278	ITALIAN	280

Note! Some of the character sets in this table correspond to the same coded page. If omitted, the code page defaults to 037.

Table 12-4 OpenVMS Character Sets

Multinational Character Sets	**National Replacement Character Sets**
DECMCS (default)	NORTH_AMERICA
LATIN1	FLEMISH CANADIAN_FRENCH BRITISH DANISH AUSTRIAN_GERMAN DUTCH ITALIAN SWISS_FRENCH SWISS_GERMAN SWEDISH NORWEGIAN BELGIAN_FRENCH SPANISH PORTUGUESE

TN3270 Keypad Graphics Characters

The TN3270 keyboard mapping key definitions permit mapping keypad graphics characters (0-9 . , -) to themselves rather than to other 3270 functions. Modify the MAP3270.DAT file if you emulate a TN3270 keyboard but want to use the graphics keypad characters as they are on OpenVMS keys.

When you modify the MAP3270.DAT file to map the graphics keypad, use the key naming conventions shown in Table 12-5. Then make the keypad map to the graphics on the keys, as follows:

1 Modify TCPWARE:MAP3270.DAT (or a variant of it) to include the entry as shown in Example 12-8.
2 Search through the file and delete any other occurrences of these escape sequences.
3 The enter key (\EOM) maps to the HOME function by default. Change it to ENTER if desired.

Table 12-5 Graphics Keypad Naming Conventions

Use...	To represent OpenVMS Keypad Key...
NUM0 (through) NUM9	graphics 0 through 9
PERIOD	period (.)
COMMA	comma (,)
HYPHEN	hyphen (-)

Example 12-8 Sample Keypad Graphics Characters Definitions in the MAP3270.DAT File

```
# Use keys on numeric keypad as themselves (numbers)
hyphen = '\EOm'; comma = '\EOl'; period = '\EOn';
num0 = '\EOp';
num1 = '\EOq'; num2 = '\EOr'; num3 = '\EOs';
num4 = '\EOt'; num5 = '\EOu'; num6 = '\EOv';
num7 = '\EOw'; num8 = '\EOx'; num9 = '\EOy';
```

TN3270 Screen Printing and Dumping

You can print or dump to a file a TN3270 session screen by using additional qualifiers with the TELNET /TN3270 or OPEN /TN3270 command. You can specify a screen print or dump either during or after opening a connection to a host.

To print a screen in TN3270 mode or dump a screen into a specified file (see Example 12-9):

1 If you want to print the ensuing TN3270 screen while opening a TN3270 host connection, specify at the DCL prompt:

```
$ TELNET host /TN3270 /PRINT=(QUEUE=qname)
```

Or specify at the `TELNET>` prompt:

```
TELNET>OPEN host /TN3270 /PRINT=(QUEUE=qname)
```

You can also add the FORM parameter, which specifies the form name for the print queue, as in:

```
TELNET>OPEN host /TN3270 /PRINT=(QUEUE=qname, FORM=form-name)
```

2 If you want to print the current TN3270 session screen when already in TN3270 mode, **Ctrl/C** out of the session and specify at the `TELNET>` prompt:

```
TELNET> SET PRINT /QUEUE=qname [/FORM=form-name]
```

The /QUEUE qualifier is like the QUEUE parameter and the optional /FORM qualifier is like the FORM parameter in step 1 previously.

3 If you want to dump the ensuing TN3270 screen into a file while opening a TN3270 host connection, specify at the DCL prompt:

```
$ TELNET host /TN3270 /PRINT=(FILE=filename)
```

Or specify at the TELNET> prompt:

```
TELNET>OPEN host /TN3270 /PRINT=(FILE=filename)
```

The default print setting is /PRINT=(FILE=SYS$LOGIN:TN3270.TXT, NOAPPEND). You can also use the APPEND keyword that appends the current screen dump onto an existing filename (NOAPPEND is the default):

```
TELNET>OPEN host /TN3270 /PRINT=(FILE=filename, [NO]APPEND)
```

4 If you want to dump the ensuing TN3270 screen into a file when already in TN3270 mode, `Ctrl/C` out of the session and specify at the TELNET> prompt:

```
TELNET>SET PRINT /FILE=filename[/[NO]APPEND]
```

The default print setting is SET PRINT /FILE=SYS$LOGIN:TN3270.TXT /NOAPPEND. The /FILE qualifier is like the FILE parameter and the optional /APPEND qualifier is like the APPEND keyword in step 3.

5 Resume the current session. When you are at the desired screen, press the "Escape" character (however it is defined) together with the character **P** (uppercase or lowercase). (In the example, the "Escape" character is defined as **F11** so that the print key sequence is **F11/P**.) Exit the session and check for the existence of the print queue or file.

Example 12-9 Printing and Dumping TN3270 Screens

```
$ TELNET LOCIS.LOC.GOV /TN3270 /PRINT=(QUEUE=ENG_PRINTER_ANSI)
L O C I S :  LIBRARY OF CONGRESS INFORMATION SYSTEM

      Choice: F11/Pquit

$ SHOW QUEUE ENG_PRINTER_ANSI
<shows active printer queue>

$ TELNET LOCIS.LOC.GOV /TN3270
L O C I S :  LIBRARY OF CONGRESS INFORMATION SYSTEM

      Choice: Ctrl/C
TELNET>SET PRINT /QUEUE=ENG_PRINTER_ASCII
TELNET>RESUME

      Choice: F11/Pquit
TELNET>QUIT
$ SHOW QUEUE ENG_PRINTER_ANSI
<shows active printer queue>
$ TELNET LOCIS.LOC.GOV /TN3270 /PRINT=(FILE=PRINTFILE.TXT, APPEND)
L O C I S :  LIBRARY OF CONGRESS INFORMATION SYSTEM

      Command ===>F11/Pquit
TELNET>QUIT
$ DIR PRINTFILE.TXT
<shows filename in directory; screen is appended onto existing file>

.TELNET>OPEN LOCIS.LOC.GOV /TN3270

      Command ===>Ctrl/C
TELNET>SET PRINT /FILE=PRINTFILE.TXT /APPEND
TELNET>RESUME

     Command ===> quit F11/P
```

```
TELNET>QUIT
$ TYPE PRINTFILE.TXT
<shows file; screen is appended onto existing file>
```

Sample Session

This section shows a sample Client-TELNET session.

See Example 12-10 for the corresponding numbered steps. In this sample session, a user on IRIS:

1 Starts TELNET.

2 Enters the SHOW STATUS command.

3 Connects to TULIP.

4 Logs in and does some work. (Note the appearance of the `PASSCODE:` prompt, since this user is protected using TCPware's Token Authentication.)

5 Enters the escape (attention) character to return to the TELNET prompt.

6 Changes the escape (attention) character and enters a SHOW STATUS command.

7 Enters the RESUME command to return to TULIP.

8 Logs out of TULIP.

9 Exits TELNET.

Example 12-10 Sample Client-TELNET Session

```
(Iris) $ TELNET
TELNET>SHOW STATUS
Client-TELNET V6.0-0 Copyright (c) Process Software
No connection established.
Terminal type list: VT300, DEC-VT300, IBM-3278-2
"^\" is the escape (attention) character

TELNET>OPEN TULIP
%TCPWARE_TELNET-I-TRYING, trying tulip.flower.com,telnet
(192.168.1.56,23)...
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^\"

SunOS UNIX 4.1 (tulip.flower.com)(ttyp2)

login: root
Password:

PASSCODE:

Last login: Wed Feb 21 10:57:25 from 198.168.1.105
Sun Microsystems Inc.   SunOS 5.9       Generic May 2002

tulip>ls
bin      mnt     notes     test.c     test_def.h
tulip>^ \

TELNET>SET ESCAPE "^A"
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^A"
TELNET>SHOW STATUS
Client-TELNET V6.0-0 Copyright (c) Process Software
Connected session:
     --1. tulip.flower.com,telnet (192.168.1.56,23).
"^A" is the escape (attention) character
TELNET>RESUME
tulip ls -A
.             .forward    bin         test.c
..            .login      mnt         test_def.h
.cshrc        .profile    notes
TELNET>EXIT
(Iris)$
```

Command Reference

The following pages consist of command descriptions for the available Client-TELNET commands.

You interact with Client-TELNET by typing commands at the `TELNET>` prompt. Client-TELNET supports the following OpenVMS-style commands:

CLOSE	SET *[*NO*]* BINARY	SET *[*NO*]*LOCAL_FLOW
DEFINE/KEY	SET *[*NO*]*BRK	SET LOG
EXIT	SET DEBUG	SET PRINT
FLUSH	SET DELETE_ALLOWED	SET TERMINAL_TYPE
HELP	SET *[*NO*]*EC	SET TRANSLATION
OPEN	SET *[*NO*]*EL	SET *[*NO*]*XDISPLOC
RESUME	SET *[*NO*]*ESCAPE	SHOW OPTIONS
SEND	SET *[*NO*]*FLUSH	SHOW STATUS
SET *[*NO*]*AO	SET *[*NO*]*FORWARD	SHOW TRANSLATION
SET *[*NO*]*AYT	SET *[*NO*]*GA	SPAWN
SET *[*NO*]*BACKWARD	SET *[*NO*]*IP	

Table 12-6 TELNET Command Synonyms

Synonym	**Equivalent**	**Synonym**	**Equivalent**
BYE or QUIT	EXIT	SET HOST	OPEN

CONNECT	OPEN	STATUS	SHOW STATUS
DISCONNECT	CLOSE	Z	SPAWN
ESCAPE	SET ESCAPE		

This command reference includes:

Name of the command	Format of the command	Qualifiers, if applicable
Synonym, if available	Parameters, if applicable	Examples of usage

CLOSE

Closes the current connection or the session specified by the session number. If you are not connected to a remote host, this command has no effect.

When you open a session using the alternate TELNET *host* format, the CLOSE command:

- Exits TELNET if the connection is the only session.
- Keeps you in TELNET with the other session(s) open if there is at least one other session.

Format

CLOSE *[session-number]*

Synonym

DISCONNECT *[session-number]*

Parameter

session-number

Session number to close, based on the session number displayed by the SHOW STATUS command. If omitted, closes the current session. If there are any other connections open, Client-TELNET resets the current session to the "next" one.

Examples

You can use the SHOW STATUS command to display a list of open connections. These examples start with HOMER as the current session. There are three TELNET connections, as follows, with the current session being on HOMER:

```
   1. BART.nene.com,  telnet (192.168.1.92,23).
   2. MARGE.nene.com, telnet (192.168.1.91,23).
-->3. HOMER.nene.com, telnet (192.168.1.90,23).
```

1 This example ends the session on MARGE. The current session is still HOMER. You can close any other session without affecting the status of the current session.

```
TELNET>CLOSE 2
%TELNET-S-LCLCLOSED, Local connection closed
-TELNET-I-SESSION, Session 02, host marge.nene.com, port 23
%TELNET-I-CURRSESSION, current session is now 3, homer.nene.com
```

2 This example ends the current session on HOMER and defaults to the session on BART. Because you are closing the current session, Client-TELNET resets the current session to the "next" connected session.

```
TELNET>CLOSE
%TELNET-S-LCLCLOSED, Local connection closed
-TELNET-I-SESSION, Session 03, host homer.nene.com, port 23
%TELNET-I-CURRSESSION, current session is now 1, bart.nene.com
```

DEFINE/KEY

Associates an equivalence string and a set of attributes with a key on the terminal keyboard.

Format

DEFINE/KEY ***key-name ["]equivalence-string["]***

Parameters

key-name

Name of the key to define. Table 12-7 lists key designations for three terminal types:

- On LK201 terminals, the numeric keypad, editing keypad (except the $ and ^ arrow keys), or function key row (except F1 through F5).
- On VT52 terminals, all definable keys are on the numeric keypad.
- On VT100-type terminals, you can also define the ⇐ and ⇒ keys. On VT200 terminals, the ⇐, ⇒, and F6 through F14 keys are for command line editing. Issue the DCL command SET TERMINAL/ NOLINE_EDITING to define these keys before you run Client-TELNET. You can also press **`Ctrl/V`** to enable keys F7 through F14.

Table 12-7 Key Designations for Three Terminal Types

Key Name	LK201	VT100-type	VT52
PF1	PF1	PF1	[blue]
PF2	PF2	PF2	[red]
PF3	PF3	PF3	[gray]
PF4	PF4	PF4	n/a
KP0,...KP9	0,...,9	0,...,9	0,...,9
PERIOD	.	.	.
COMMA	,	,	,
MINUS	-	-	-
ENTER	Enter	ENTER	ENTER
LEFT	‹	‹	‹
RIGHT	fi	fi	fi

Find (E1)	Find		
Insert_Here (E2)	Insert_Here		
Remove (E3)	Remove		
Select (E4)	Select		
Prev_Screen (E5)	Prev_Screen		
Next_Screen (E6)	Next_Screen		
HELP	Help		
DO	Do		
F6,...,F20	F6,...,F20		

equivalence-string

String to substitute when you press the key. If the string contains spaces, enclose it in quotes.

Qualifiers

/ECHO (default**)**
/NOECHO

/ECHO (the default) displays the equivalence string on your screen after you press the key.
/NOECHO disables this. Use /NOECHO with /TERMINATE only.

/IF_STATE=*(state-name[,state-name,...])*
/NOIF_STATE (default)

/IF_STATE specifies one or more *state-names* (alphanumeric strings separated by commas) for the key definition to be in effect. You can omit the parentheses if you specify only one *state-name*.
/NOIF_STATE is the default, where the current state applies.

Establish states using the /SET_STATE qualifier (see below). If you specify several *state-names*, you can define a key to have the same function in all the specified states.

/LOCK_STATE
/NOLOCK_STATE (default)

/LOCK_STATE specifies that the state set by the /SET_STATE qualifier remains in effect until explicitly changed. /NOLOCK_STATE is the default, where the state set by /SET_STATE is in effect only for the next definable key that you press or for the next read terminating character that you type.

You can only specify /LOCK_STATE with /SET_STATE.

/**SET_STATE**=*state-name*
/NOSET_STATE (default)

/SET_STATE specifies the *state-name* (an alphanumeric string) to set when pressing the key. *State-name* is an alphanumeric string. The default is /NOSET_STATE, where the current locked state, if any, remains in effect.

/**TERMINATE**
/NOTERMINATE (default)

Specifies whether to terminate (execute) the current equivalence string when you press the key. /NOTERMINATE (the default) lets you create key definitions that insert text into command lines, at prompts, or into other text you type.

EXIT

Exits the Client-TELNET utility and returns to the DCL level.

If there is an open connection or log file, Client-TELNET closes it before exiting.

Once you exit, all connections to remote hosts are disconnected.

Format

EXIT

Synonyms

QUIT

BYE

`Ctrl/Z`

FLUSH

Discards all characters currently in the output stream from the server.

Ignored if no connection is open.

Note! Unlike the flush character (see theSET command), the FLUSH command does not use the timing-mark option.

Format

FLUSH

HELP

Obtains help on using the Client-TELNET utility.

TELNET help uses the OpenVMS interactive help facility.

To exit the help facility, press the RETURN key until you return to the `TELNET>` prompt.

Format

HELP *[topic]*

Parameter

topic

Topic on which you want help. Optional.

OPEN

Opens a connection to a remote host. You can open up to ten connections at any one time. The connection remains open until you log out of the remote host, or use the CLOSE or EXIT command at the `TELNET>` prompt.

To use Kerberos version 4 authentication with TELNET, you must first get a ticket-granting ticket (TGT) from the Kerberos Server. (See Chapter 4, *Kerberos User Commands*.)

If you are designated by the system administrator as having password authentication through Token Authentication, you need to enter the PASSCODE in addition to the username and password at a separate `PASSCODE:` prompt (see Setting Up a Startup Command File). Depending on which type of SecurID card you were assigned:

- Enter a combination of your memorized personal identification number (PIN) and the tokencode that appears on the card (with no separating space) at the `PASSCODE:` prompt, or
- Enter your memorized PIN on the PINPAD™ card and the resulting tokencode that appears on the card at the `PASSCODE:` prompt.

See Chapter 14, *Token Authentication: Protecting Logins*, for details on obtaining PASSCODEs.

Note! The same parameters and qualifiers apply to the TELNET command on the DCL level as apply to the OPEN command within TELNET.

Format

OPEN *[host [port]]*

Synonyms

CONNECT*[host [port]]*
SET HOST*[host [port]]*

Parameters

host

Name of the remote host to which you want to connect. The host must exist on the network.

Enter OPEN *host* to open a remote connection and start the login sequence, if any. If you omit *host* and a connection is open, Client-TELNET resumes the session to that host.

port

Nonstandard service name or number of the remote port to which you want to connect. The default is **`TELNET`** or **`23`** (for the TELNET Server). Use only to connect to a nonstandard server. ALTERNATIVE: Use the /PORT qualifier (DO NOT use both in the same command; see Example 6).

Qualifiers

/AUTHENTICATION*[=auth-type]*

Determines the authentication method. If *auth-type* is **`KERBV4`** (or the value is omitted), Kerberos version 4 authentication is used. If *auth-type* is **`NULL`** (or the entire qualifier is omitted), standard authentication is used.

/CREATE *[*=(PERMANENT, BROKE_TIMO=*seconds*, CLOSE_DASSGN, INTERVAL=*seconds*, *[*NO*]*KEEPALIVE, NOOPCOM, NOTCONNECTED_OK, RETRIES=*number*), SHUT_ABORT) *]*

Associates the local client end of the TELNET connection to an NTA device. Lets you use the connection for terminal activities such as printing or running applications. Supports /RAW,
/LOGICAL, and /TIMEOUT.

The /CREATE keyword creates the NTA device as pre-allocated so that it is not deleted when exiting TELNET. However, deallocating the device deletes it automatically when there are no process channels assigned to it (the reference count drops to zero). The **PERMANENT** keyword causes the client NTA device NOT to be deleted automatically when there are no process channels assigned to it, thus creating a permanent connection similar to an application LTA device for LAT. As with LAT, if the TELNET connection is broken, the Client-TELNET device tries to reconnect to the specified host and port. Further parameters control the broken connection and reconnection algorithms:

BROKE_TIMO=*seconds*	Used to determine when a connection is broken. (Note that the OPEN /TIMEOUT qualifier value is used in establishing the connection, and another timeout of eight minutes is used when sending data.) If omitted, the /TIMEOUT value is used. Also applies to non-permanent NTA devices (when using OPEN/CREATE without the PERMANENT keyword).
CLOSE_DASSGN	Specifies that when the last channel is deassigned from the NTA device, the underlying TCP connection is closed. The default is NOT to close the TCP connection. Use with the PERMANENT keyword only.
INTERVAL=*seconds*	Connection retry interval, the minimum time to wait until another connect is attempted. The default is `120` seconds (two minutes). Use with the PERMANENT keyword only.
KEEPALIVE or **NOKEEPALIVE**	Controls whether keep-alive segments are sent to the remote port. The default is KEEPALIVE. Also applies to non-permanent NTA devices (when using OPEN/CREATE without the PERMANENT keyword).
NOOPCOM	Specifies that no OPCOM messages are used when a permanent NTA device fails to reconnect or reconnects after an initial failure. OPCOM messages are sent by default.
NOTCONNECTED_OK	A permanent NTA device is created even if a TCP connection cannot initially be set up.
RETRIES=*number*	Number of times to try to reconnect after a connection breaks; the default is `-1`, handled as an unsigned number and thus actually 4,294,967,295, which is, in effect, infinite. Use with the PERMANENT keyword only.

SHUT_ABORT	Specifies that a permanent NTA device will do extra TCP device cleanup after the underlying TCP connection is shutdown. This is similar to doing NETCU> KILL CONNECTION for a closed TCP device.

Setting RETRIES to 0 means that when either end closes the TCP connection, no reconnects automatically occur. However, a reconnection attempt is made without delay when a write operation to the permanent NTA device occurs. If RETRIES is not set to 0, automatic retries occur when the connection closes. If all those retries fail, and a write is done later to the NTA device, then the specified number of retries is attempted.

Here is a typical command to create a TELNET connection to a printer (note the use of /RAW to avoid sending TELNET options negotiation data):

```
TELNET /RAW /CREATE=(PERM, RETRIES=0, CLOSE) host port
```

After TELNET creates a permanent NTA device with an underlying TCP connection, the NTA device's reference count drops to 0; thus the TCP connection is closed. When a write operation occurs to the NTA device, an attempt is made to re-establish the TCP connection. Meanwhile the data being written is held so that it can be sent when reconnected. If all reconnects fail, the write data is dropped. When the application deassigns its channels to the NTA device, its TCP connection is again closed.

To specify that the permanent NT device should be treated as a local terminal rather than a remote terminal (to allow for spooling of the device), add the 'local' keyword to the TELNET “create” qualifier:

```
TELNET /CREATE=(PERM,LOCAL)
```

/HOST_CHARACTER_SET=*name*

Use with the /TN3270 qualifier to set the national EBCDIC character set for TN3270 Internationalization. Table 12-3 shows the supported character sets and their corresponding IBM code page numbers.

/**LOGICAL**=*name [*/TABLE=*table][*/MODE=*mode]*

Logical name defined for the allocated NTA device. Use only with the /CREATE qualifier. The *table* values are **PROCESS** (the default), **JOB, GROUP,** or **SYSTEM**. The *mode* values are **SUPERVISOR** (the default) or **EXECUTIVE**.

/PORT=*port*

Nonstandard service name or number of the remote port to which you want to connect. The default is 23 (for the TELNET Server). Use only to connect to a nonstandard server. ALTERNATIVE: Use the *port* command parameter (DO NOT use both in the same command; see Set).

/PRINT=*[(]*{FILE=*file[*, *[*NO*]*APPEND*]*} | {QUEUE=*qname[*, FORM=*form]*}*[)]*

Prints a TN3270 screen or dumps it into a file. Use only with the /TN3270 qualifier. Provides the functionality of the PRINT key, which the TCPWARE:MAP3270.DAT file defines by default as follows:

```
lprt = '\Ep' | '\EP'; # ESCAPE-p, ESCAPE-P
```

Use either FILE or QUEUE, but not both:

FILE=*file*	Output file (the default is SYS$LOGIN:TN3270.TXT). APPEND appends each print page onto the file; NOAPPEND (the default) creates a new file for each page.

QUEUE=*qname*	Location of the print queue. FORM=*form* specifies the form to use when sending the page output to a print queue.

/RAW

Specifies a raw, binary connection that does not adhere to the TELNET protocol. Use only with the /CREATE qualifier.

/REALM=*realm*

Assigns the name of the Kerberos realm. If the Kerberos Server resides in a different realm than the local host, use this qualifier. Use with the /AUTHENTICATION=KERBV4 qualifier and value. The realm is converted to lowercase unless you enclose it in quotes.

/TERMINAL_CHARACTER_SET=*name*

Use with the /TN3270 qualifier to set the OpenVMS terminal character set for TN3270 Internationalization. Table 12-4 shows the supported Multinational and National Replacement character set values.

/TIMEOUT=*seconds*

Timeout time for establishing the TELNET control connection. If not specified, the default value of **`120`** seconds (2 minutes) applies. The minimum allowable value is **`20`**.

/TN3270
/NOTN3270

/TN3270 enables TN3270 mode. Use this qualifier when you want your OpenVMS terminal to emulate an IBM 3270-class terminal but the server cannot negotiate this mode automatically. (If the server can negotiate TN3270 mode automatically, you can omit this qualifier.) Only one TN3270 session can be open at any one time. Use the /PRINT qualifier for printing or file-dumping a TN3270 screen.

Use /TN3270 with the /HOST_CHARACTER_SET and /TERMINAL_CHARACTER_SET qualifiers to support TN3270 Internationalization.

/NOTN3270 disables TN3270 mode. Use this qualifier if you connect to a remote terminal that supports both IBM 3270 mode and non-IBM 3270 connections.

Examples

1 This example opens a connection to host DAISY and enables TN3270 mode. Use /TN3270 only if the server cannot negotiate TN3270 mode automatically. Client-TELNET allows only one TN3270 session at any one time.

```
TELNET>OPEN /TN3270 DAISY

<login procedure to daisy....>

(daisy)$ Ctrl/\
TELNET>OPEN /TN3270 ROSE
%TCPWARE_TELNET-E-MAXTN3270, only one TN3270 session may be open at any time
TELNET>
```

2 This example opens a connection to host DAISY in TN3270 mode and specifies a Danish TN3270 Internationalization host character set:

```
TELNET>OPEN /TN3270 /HOST_CHARACTER_SET=DANISH DAISY

<login procedure to daisy....>
```

```
(daisy)$
```

3 This example opens three sessions. The first two use Kerberos version 4 authentication; the third uses standard authentication. The Kerberos Server realm is determined by the contents of the TCPWARE:KRB.REALMS file.

```
TELNET>OPEN /AUTH=KERBV4 BART

(bart)$ ^\
TELNET>OPEN /AUTH MARGE
(marge)$ ^\
TELNET>OPEN LISA

<login procedure to LISA....>

(lisa)$
```

4 This example opens a TN3270 connection and prints the next screen that appears to the print queue ENG_PRINTER_ASCII:

```
TELNET>OPEN DAISY /TN3270 /PRINT=(QUEUE=ENG_PRINTER_ASCII)

<login procedure to daisy....>
```

5 This example creates a permanent NTA device for the connection to MARGE port 7 for the user application. In case the connection goes down, it is set up so that automatic reconnection retries occur every 10 seconds for a total of 10 retries.

```
TELNET>OPEN /LOGICAL=MY_PORT -
_TELNET>/CREATE=(PERMANENT,INTERVAL=10,RETRIES=10) MARGE 7
%TCPWARE_TELNET-I-CREATED, _NTA2: created
$ @MY_APPLICATION MY_PORT
```

6 This example displays the results of using the *port* parameter value (telnet) together with the /PORT qualifier and value in a single command:

```
TELNET>OPEN DAISY TELNET /PORT=23
%TCPWARE_TELNET-W-CONFLICT illegal combination of command elements - check
documentation
```

7 This example displays a login session to DAISY that uses Token Authentication for password protection:

```
TELNET>OPEN DAISY
%TCPWARE_TELNET-I-TRYING, trying
DAISY.nene.com,telnet (192.168.142.7,23) ...
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^\"

           ** AUTHORIZED USE ONLY **   PHI (VAX/VMS V5.-2)

Username: PETER
Password:

Enter PASSCODE:
PASSCODE Accepted
```

RESUME

Resumes the current connection if you do not specify a session number. If you specify a session number, resumes the connection associated with the session number, as displayed by the SHOW STATUS (or STATUS) command.

Format

RESUME *[session-number]*

Parameter

session-number

Session number to resume, based on the session number the SHOW STATUS command displays. If omitted, resumes the current connection.

Examples

1 This example resumes the session on BART. Client-TELNET does not display a message if the user resumes the current session:

```
TELNET>SHOW STATUS
Connected session:
-->1. BART.humor.com, telnet (192.168.1.92,23).

TELNET>RESUME
(bart)$
```

2 This example resumes session 2 on MARGE:

```
TELNET>STATUS
Connected sessions:
   1. BART.humor.com,  telnet (192.166.1.92,23).
   2. MARGE.humor.com, telnet (192.166.1.91,23).
-->3. HOMER.illiad.com, telnet (192.162.1.90,23).

TELNET>RESUME 2
%TCPWARE_TELNET-I-RESUME, resuming session 2, MARGE.humor.com
(marge)$
```

SEND

Sends TELNET control functions or option negotiations to a remote host.

Format

SEND {*control-function* | {*command option*}}

Parameters

control-function

Table 12-8 lists the available TELNET control functions. Send a control function to gain access to functions of the remote host that are not available from the keyboard.

Table 12-8 TELNET Control Functions

Control Function	Definition
AO	Abort Output
AYT	Are You There
BACKWARD	Sends the current Client-TELNET Backward character
BRK	Break
EC	Erase Character
EL	Erase Line
ESCAPE	Sends the current Client-TELNET Escape character
FORWARD	Sends the current Client-TELNET Forward character
GA	Go-Ahead
IP	Interrupt Process
NOIP	Do Not Interrupt Process
SYNCH	SYNCH signal

Command

One of the following TELNET protocol commands used in options negotiation:

DO	WILL	DONT	WONT

Option

Negotiated TELNET option. Client-TELNET supports the following *option* keywords:

ECHO	for the ECHO option SEND WILL ECHO is an invalid command. Client-TELNET does not allow the user to send this option negotiation to the TELNET Server.
BINARY, or TRANSMIT_BINARY	for the TRANSMIT-BINARY option
SGA, or SUPPRESS_GO_AHEAD	for the SUPPRESS-GO-AHEAD option

SET *[*NO*]*AO

Defines, changes, or disables the "abort output" (AO) character. During a TELNET session, if you enter the defined AO character, Client-TELNET sends the TELNET AO control function to the server instead of the actual character.

Ignored if TN3270 mode is active.

Format

SET AO *char*
SET NOAO

Parameter

char

When entered, this character sends the TELNET AO control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default AO character. Define the initial AO character using the TCPWARE_TELNET_AO logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_AO 15`
- `$ DEFINE/PROCESS TCPWARE_TELNET_AO """^O"""`

Both commands set the AO character to `Ctrl/O` (ASCII 15). They are equivalent.

Qualifiers

/FLUSH (default)
/NOFLUSH

If you specify /FLUSH, Client-TELNET discards all characters currently in the output stream from the server when sending the AO control function. Client-TELNET uses the TELNET timing-mark option to accomplish this (the Server does not have to support this option for this feature to work). If you specify /NOFLUSH, Client-TELNET sends only the AO control function. If you omit both, the previous setting remains. The initial default is /FLUSH.

If there is no response to the timing-mark option, Client-TELNET may continue to discard output from the server. Use the FLUSH command to resume normal operation.

/**SYNCH**
/NOSYNCH (default)

Sends the AO command followed by the SYNCH signal.

Examples

1 Each of these equivalent commands sets the AO character to **`Ctrl/O`** (ASCII 15):

```
TELNET>SET AO "^O"
TELNET>SET AO 15
```

2 This example removes the previous character definition, if any, for the AO control function:

```
TELNET> SET NOAO
```

SET *[*NO*]*AYT

Defines, changes, or disables the "are you there" (AYT) character. If you enter the defined AYT character during a TELNET session, Client-TELNET sends the TELNET AYT control function to the server instead of the actual character. Ignored if TN3270 mode is active.

Format

SET AYT *char*
SET NOAYT

Parameter

char

When entered, this character sends the TELNET AYT control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default AYT character. Define the initial AYT character using the TCPWARE_TELNET_AYT logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_AYT 7`
- `$ DEFINE/PROCESS TCPWARE_TELNET_AYT """^G"""`

Both commands set the AYT character to `Ctrl/G` (ASCII 7). They are equivalent.

Qualifiers

/SYNCH
/NOSYNCH (default)

Sends the AYT command followed by the SYNCH signal.

Examples

1 Each of these equivalent commands sets the AYT character to `Ctrl/G` (ASCII 7):

```
TELNET>SET AYT "^G"
TELNET>SET AYT 7
```

2 This example removes the previous character definition, if any, for the AYT control function:

```
TELNET>SET NOAYT
```

SET *[NO]*BACKWARD

Defines, changes, or disables the "backward (one session)" (BACKWARD) character. If you enter the BACKWARD character during a TELNET session, the "previous" numbered session becomes active. The previous numbered session is the session with the next lowest session number than the current session.

If the current session already has the lowest session number, the session with the highest session number becomes active. If there is only one active session available, that session remains active. In this case SET BACKWARD has no effect.

Ignored if TN3270 mode is active.

Format

SET BACKWARD char
SET NOBACKWARD

Parameter

char

When entered, this character causes the "previous" numbered session to become active. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default BACKWARD character. Define the initial BACKWARD character using the TCPWARE_TELNET_BACKWARD logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_BACKWARD 2`
- `$ DEFINE/PROCESS TCPWARE_TELNET_BACKWARD """^B"""`

Both commands set the BACKWARD character to **`Ctrl/B`** (ASCII 2). They are equivalent.

Examples

1 Each of these equivalent commands sets the BACKWARD character to **`Ctrl/B`** (ASCII 2):

```
TELNET>SET BACKWARD "^B"
TELNET>SET BACKWARD 2
```

2 This example removes the previous character definition, if any, for the BACKWARD control function:

```
TELNET>SET NOBACKWARD
```

SET *[*NO*]*BINARY

Initiates negotiations to enable the TRANSMIT BINARY option for the client and server. This command:

- Pertains only to the current session.
- Automatically resumes the current session.

Use the SET NOBINARY command to initiate negotiations to disable the TRANSMIT BINARY option for the client and server.

Format

SET BINARY
SET NOBINARY

SET *[NO]*BRK

Defines, changes, or disables the break (BRK) character. If you define the BRK character during a TELNET session, Client-TELNET sends the TELNET BRK control function to the server instead of the actual character.

Ignored if TN3270 mode is active. The Server ignores the break character.

Format

SET BRK *char*
SET NOBRK

Parameter

char

When entered, this character sends the TELNET break control function to the server. Specified in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default BRK character. Define the initial BRK character using the TCPWARE_TELNET_BRK logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_BRK 29`
- `$ DEFINE/PROCESS TCPWARE_TELNET_BRK """^]"""`

Both commands set the break character to `Ctrl/]` (ASCII 29). They are equivalent.

Qualifiers

/FLUSH (default)
/**NOFLUSH**

If you specify /FLUSH, Client-TELNET discards all characters currently in the output stream from the server when sending the BRK function. Client-TELNET uses the TELNET timing-mark option to accomplish this (the server does not have to support this option for this feature to work).

If you specify /NOFLUSH, Client-TELNET sends only the BRK function. If you omit both, the previous setting remains. The initial default is /FLUSH.

Note! If a server fails to respond properly to the timing-mark option, Client-TELNET may continue to discard output from the server. In this case, use the FLUSH command to resume normal operation.

Examples

1 Each of these equivalent commands sets the break character to `Ctrl/]` (ASCII 29):

```
TELNET>SET BRK "^]"
TELNET>SET BRK 29
```

2 This example removes the previous character definition, if any, for the break control function:

```
TELNET>SET NOBRK
```

SET DEBUG

Enables or disables the display of debugging information.

Format

SET DEBUG /CLASS=*[(]keyword[,...)]*

Qualifier

/CLASS[=keyword]

SET DEBUG requires the /CLASS qualifier. The optional *keyword* specifies the classes of debugging information to enable or disable. Use parentheses for multiple keywords separated by commas. Table 12-9 lists the supported keywords.

Table 12-9 Class Keywords

Keyword	**Description**
ALL	Enables the display of all classes.
OPTIONS	Enables the display of options negotiation information. Client-TELNET displays messages when it sends or receives TELNET options.
NETINPUT	Logs data that Client-TELNET receives and sends while in TN3270 mode.
NETOUTPUT	Logs data that Client-TELNET sends while in TN3270 mode.
NONE	Disables the display of all classes.
TTYINPUT	Logs data entered by the user at the terminal.

The initial setting is NONE.

SET DEBUG alone, or SET DEBUG /CLASS without the keyword, shows the current debug classes.

Examples

1 This example enables the display of options negotiation information:

```
TELNET>SET DEBUG/CLASS=OPTIONS
```

2 This example enables the display of options negotiation information and log-data sent and received while in TN3270 mode:

```
TELNET>SET DEBUG/CLASS=(OPTIONS, NETINPUT)
```

SET DELETE_ALLOWED

Allows deletion of an NTA device originally set up as permanent. The deletion occurs when there are no process channels assigned to the device.

See the OPEN /CREATE command for details on creating permanent NTA devices.

Format

SET DELETE_ALLOWED *nta-device*

Parameter

nta-device

NTA device set up using OPEN /CREATE=(PERMANENT...).

Example

This example allows the NTA33: device to be deleted when no channels are assigned to it:

```
TELNET>SET DELETE NTA33:
```

SET *[NO]*EC

Defines, changes, or disables the "erase character" (EC) character. If you enter the defined EC character during a TELNET session, Client-TELNET sends the TELNET EC control function to the server instead of the actual character.

Ignored if TN3270 mode is active.

Format

SET EC *char*
SET [NO]EC

Parameter

char

When entered, this character sends the TELNET EC control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default EC character. Define the initial EC character using the TCPWARE_TELNET_EC logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_EC 4`
- `$ DEFINE/PROCESS TCPWARE_TELNET_EC """^D"""`

Both commands set the EC character to `Ctrl/D` (ASCII 4). They are equivalent.

Examples

1 Each of these equivalent commands sets the EC character to `Ctrl/D` (ASCII 4):

```
TELNET>SET EC “^D”
TELNET>SET EC 4
```

2 This example removes the previous character definition, if any, for the EC control function:

```
TELNET>SET NOEC
```

SET *[NO]*EL

Defines, changes, or disables the "erase line" (EL) character. If you enter the defined EL character during a TELNET session, Client-TELNET sends the TELNET EL control function to the server instead of the actual character.

Ignored if TN3270 mode is active.

Format

SET EL *char*
SET NOEL

Parameter

char

When entered, this character sends the TELNET EL control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default EL character. Define the initial EL character using the TCPWARE_TELNET_EL logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_EL 21`
- `$ DEFINE/PROCESS TCPWARE_TELNET_EL """^U"""`

Both commands set the EL character to `Ctrl/U` (ASCII 21). They are equivalent.

Examples

1 Each of these equivalent commands sets the EL character to `Ctrl/U` (ASCII 21):

```
TELNET>SET EL "^U"
TELNET> SET EL 21
```

2 This example removes the previous character definition, if any, for the EL control function:

```
TELNET>SET NOEL
```

SET *[NO]*ESCAPE

SET ESCAPE changes the escape (attention) character. This command allows you to change the character to a key that is more convenient. The default escape character is ^\. You may want to change the escape character if the remote host uses that character to perform some function or if your terminal cannot generate the character.

SET NOESCAPE disables the escape (attention) character.

SET ESCAPE is ignored if TN3270 mode is active. However, SET NOESCAPE applies to all sessions, including TN3270 sessions.

Format

SET ESCAPE *char*
SET NOESCAPE

Synonym

ESCAPE = SET ESCAPE

Parameter

char

You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

You can redefine the default escape (attention) character by defining the logical TCPWARE_TELNET_ESCAPE (in the process, job, group, or system logical name tables). The logical value has the same syntax as *char*. To define it, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_ESCAPE 24`
- `$ DEFINE/PROCESS TCPWARE_TELNET_ESCAPE """^X"""`

Both commands set the escape character to ASCII code 24 (**`Ctrl/X`**). They are equivalent.

- `$ DEFINE/SYSTEM/EXEC TCPWARE_TELNET_ESCAPE -1`

The `-1` value disables the escape (attention) character.

Examples

1 Each of these equivalent commands sets the escape character to **`Ctrl/X`** (ASCII 24):

```
TELNET>SET ESCAPE "^X"
TELNET>SET ESCAPE 24
```

2 This example sets the escape character to right brace (}):

```
TELNET>SET ESCAPE "}"
```

3 This example removes the previous escape (attention) character definition, if any:

```
TELNET>SET NOESCAPE
```

SET *[NO]*FLUSH

Defines, changes, or disables the flush character.

If you enter the defined flush character during a TELNET session, Client-TELNET discards all characters currently in the output stream from the server. Client-TELNET uses the TELNET timing-mark option to accomplish this (a TELNET server need not support this option for this feature to work).

Note! Client-TELNET ignores SET FLUSH when TN3270 mode is active.
If a Server fails to respond properly to the timing-mark option, Client-TELNET may continue to discard all output from the server. In this case, use the FLUSH command to resume normal operation.

Format

SET FLUSH *char*
SET NOFLUSH

Parameter

char

When entered, this character discards all characters currently in the output stream from the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default flush character. Define the initial flush character using the TCPWARE_TELNET_FLUSH logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_FLUSH 15`
- `$ DEFINE/PROCESS TCPWARE_TELNET_FLUSH """^O"""`

Both commands set the flush character to `Ctrl/O` (ASCII 15). They are equivalent.

Examples

1 Each of these equivalent commands sets the flush character to `Ctrl/O` (ASCII 15):

```
TELNET>SET FLUSH "^O"
TELNET>SET FLUSH 15
```

2 Removes the previous character definition, if any, for the flush feature.

```
TELNET> SET NOFLUSH
```

SET *[NO]*FORWARD

Defines, changes, or disables the "forward [one session]" (FORWARD) character. If you enter the defined FORWARD character during a TELNET session, the "next" numbered session becomes active. The next numbered session is the session with the next highest session number than the current session.

If the current session already has the highest session number, the session with the lowest session number becomes active. If there is only one active session available, that session remains active. In this case SET FORWARD has no effect.

Ignored if TN3270 mode is active.

Format

SET FORWARD *char*
SET NOFORWARD

Parameter

char

When entered, this character causes the "next" numbered session to become active. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default FORWARD character. Define the initial FORWARD character using the TCPWARE_TELNET_FORWARD logical name (in the process, job, group, or system logical name tables). This logical value has the same syntax as char. To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_FORWARD 1`
- `$ DEFINE/PROCESS TCPWARE_TELNET_FORWARD """^A"""`

Both commands set the FORWARD character to `Ctrl/A` (ASCII 1). They are equivalent.

Examples

1 Each of these equivalent commands sets the FORWARD character to `Ctrl/A` (ASCII 1):

```
TELNET>SET FORWARD "^A"
TELNET>SET FORWARD 1
```

2 This example removes the previous character definition, if any, for the FORWARD control function:

```
TELNET>SET NOFORWARD
```

SET *[*NO*]*GA

Defines, changes, or disables the "go-ahead" (GA) character. If you enter the defined GA character during a TELNET session, Client-TELNET sends the TELNET GA control function to the server instead of the actual character.

Ignored if TN3270 mode is active.

Format

SET GA *char*
SET NOGA

Parameter

char

When entered, this character sends the TELNET GA control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default GA character. Define the initial GA character using the TCPWARE_TELNET_GA logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_GA 9`
- `$ DEFINE/PROCESS TCPWARE_TELNET_GA """^I"""`

Both commands set the GA character to **`Ctrl/A`** (ASCII 9). They are equivalent.

Examples

1 Each of these equivalent commands sets the GA character to **`Ctrl/A`** (ASCII 9):

```
TELNET>SET GA "^I"
TELNET>SET GA 9
```

2 This example removes the previous character definition, if any, for the GA control function:

```
TELNET> SET NOGA
```

SET *[*NO*]*IP

Defines, changes, or disables the "interrupt process" (IP) character. If you enter the defined IP character during a TELNET session, Client-TELNET sends the TELNET IP control function to the server instead of the actual character.

Ignored if TN3270 mode is active.

Format

SET IP *char*
SET NOIP

Parameter

char

When entered, this character sends the TELNET IP control function to the server. You can specify this character in either of the following formats:

Numeric	ASCII value of the character.
String	Character string enclosed in quotes. Specify control characters by typing a caret (^) before the character.

There is no default IP character. Define the initial IP character using the TCPWARE_TELNET_IP logical name (in the process, job, group, or system logical name tables). To define the logical, use one of the following formats:

- `$ DEFINE/PROCESS TCPWARE_TELNET_IP 25`
- `$ DEFINE/PROCESS TCPWARE_TELNET_IP """^Y"""`

Both commands set the IP character to **`Ctrl/Y`** (ASCII 25). They are equivalent.

Qualifiers

/FLUSH (default)
/NOFLUSH

With /FLUSH, Client-TELNET discards all characters currently in the server's output stream when sending the IP control function. It uses the TELNET timing-mark option (the server does not have to support this option for this feature to work). With /NOFLUSH, Client-TELNET sends only the IP control function. If you omit both, the previous setting remains. The initial default is /FLUSH.

If a Server fails to respond properly to the timing-mark option, Client-TELNET can continue to discard all output from the server. If so, use FLUSH to resume normal operation.

/**SYNCH**
/NOSYNCH (default)

Sends the IP command followed by the SYNCH signal.

Examples

1 Each of these equivalent commands sets the IP character to **`Ctrl/Y`** (ASCII 25):

```
TELNET>SET IP "^Y"
TELNET>SET IP 25
```

2 This example removes the previous character definition, if any, for the IP control function.

```
TELNET>SET NOIP
```

SET *[*NO*]*LOCAL_FLOW_CONTROL

Controls the handling of the XON/XOFF characters (**`Ctrl/S`** and **`Ctrl/Q`**) when connected to a remote system. **`Ctrl/S`** stops transmission and **`Ctrl/Q`** resumes TELNET transmission. Under normal conditions, the terminal driver processes **`Ctrl/S`** and **`Ctrl/Q`** locally and does not send them to the remote TELNET server.

Client-TELNET supports RFC 1372 (*Telnet Remote Flow Control Option*), which lets the remote server tell the client when to enable and disable local flow control. These commands are not related to that option, but rather let the user control the local flow control setting if the remote server does NOT support the Remote Flow Control Option.

Use SET NOLOCAL_FLOW_CONTROL to pass the **`Ctrl/S`** and **`Ctrl/Q`** characters to the remote TELNET server and NOT process them locally.

The default flow control setting depends on the TT$V_TTSYNC value for the terminal. You can set "TTSync" mode (local flow control) outside of TELNET by using the DCL SET TERMINAL /TTSYNC command, or set "No TTSync" mode (server flow control) by using the DCL SET TERMINAL /NOTTSYNC command; some full-screen editors also set these modes. However, if you are inside TELNET, SET NOLOCAL_FLOW_CONTROL can force the terminal into "No TTSync" mode for a particular connection.

Format

SET LOCAL_FLOW_CONTROL (default)
SET NOLOCAL_FLOW_CONTROL

Example

```
TELNET>SET NOLOCAL
TELNET>SHOW STATUS
Client-TELNET  V6.0-0  Copyright (c) Process Software

Connected session: -->1. beans.example.edu, telnet (192.168.0.50)

Terminal type:  VT300
Local flow control: OFF
"^D" is the escape (attention) character.
```

SET LOG

Opens or closes a log file. Client-TELNET uses a log file to save the output from a remote host. While connected to a remote host, Client-TELNET also puts all output the remote host sends your terminal into the log file.

SET LOG logs output from every connected session. If multiple connections exist, there is no way to specify that you want to log only output from a specified session to the log file.

Format

SET LOG *[file]*

Opens the local file *file* and begins logging.

To close a log file (and stop logging), enter SET LOG with no file specification.

Parameter

file

OpenVMS file specification of the file that logs the remote host's output. If omitted, Client-TELNET closes the present log file (if there is one).

Qualifiers

/DATA (default)
/**NODATA**

/DATA logs all data sent to the specified file (the default). /NODATA disables this.

/**OPTIONS**
/NOOPTIONS (default)

/OPTIONS prints option negotiations to the specified log file, in addition to performing normal logging. /NOOPTIONS (the default) disables options printing.

Examples

1 This example opens the file TEXT.LOG and enables logging:

```
TELNET>SET LOG TEXT.LOG
```

2 This example closes a log file and stops logging:

```
TELNET>SET LOG
```

3 This example opens the file TEXT.LOG, enables normal logging, and prints options negotiations to the TEXT.LOG file:

```
TELNET>SET LOG TEXT.LOG /OPTIONS
```

4 This example opens the file TEXT.LOG and prints only option negotiations (and no data) to the TEXT.LOG file:

```
TELNET>SET LOG TEXT.LOG /OPTIONS /NODATA
```

SET PRINT

Sets how you want the PRINT key to work while in TN3270 mode. You must be in TN3270 mode to use this command.

If you omit the qualifiers, the default is SET PRINT /FILE=SYS$LOGIN:TN3270.TXT /NOAPPEND.

This means that the default print setting is OPEN/TN3270 /PRINT=(FILE=SYS$LOGIN:TN3270.TXT, NOAPPEND).

Format

SET PRINT*[qualifiers]*

Opens local file *file* and begins logging.

To close a log file (and stop logging), enter SET LOG with no file specification.

Qualifiers

/**APPEND**
/NOAPPEND (default)

Use with the /FILE qualifier only. /APPEND appends the TN3270 screen dump onto the specified file. /NOAPPEND creates a new file or overwrites the existing one.

/FILE=*filename*

File in which to dump the TN3270 screen. You can use this with the optional /APPEND or /NOAPPEND qualifier.

/FORM=*form-name*

Use with the /QUEUE qualifier only. Specifies the form name to use in a TN3270 screen print.

/QUEUE=*qname*

Queue to which to print the TN3270 screen. You can use this with the optional /FORM qualifier and value.

Examples

1 This example sets the print behavior so that it prints the current TN3270 screen to a print file and appends it onto the end of the file:

```
TELNET>SET PRINT /FILE=PRINTFILE.TXT /APPEND
```

2 This example sets the print behavior so that it prints the current TN3270 screen to a print queue:

```
TELNET>SET PRINT /QUEUE=ENG_PRINTER_ASCII
```

SET TERMINAL_TYPE

Requests the server to support a specific terminal type or types if negotiating the terminal type option.

Normally, you do not need to use this command. Client-TELNET uses the following default list of supported terminal types: VT52, VT55, VT61, VT62, VT100, VT102, VT125, VT131, VT132, VT200, VT220, VT240, VT300, VT320, VT340, and IBM-3278-*model-number*.

If you specify an IBM-3278 terminal type, make sure your local terminal supports the screen size associated with the specified model number. If your terminal does not support the screen size, the data will not display properly.

See Supported I for screen sizes for each model.

Use the SHOW STATUS or SHOW OPTIONS commands to show the current terminal type used.

The TCPWARE_TELNET_TERMINAL_TYPE logical performs the same function as the SET TERMINAL_TYPE command. This logical requires the following syntax:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_TELNET_TERMINAL_TYPE "type"
```

Format

SET TERMINAL_TYPE *type[,type,...]*

Parameter

type

A valid terminal type. Client-TELNET requests the server to support these types in the specified order.

Examples

1 This example requests the server to support the VT300 and VT100 terminal types, in that order:

```
TELNET>SET TERMINAL_TYPE VT300, VT100
```

2 This example requests the server to support the IBM-3278-3 terminal type. If possible, Client-TELNET resizes the local window to accommodate a 32 x 80 screen size for model 3 (see Supported I).

```
TELNET>SET TERMINAL_TYPE IBM-3278-3
```

SET TRANSLATION

Sets the carriage return/line feed (CR/LF) character translation.

Does not apply to TN3270 mode.

Format

SET TRANSLATION

Qualifiers

/RECEIVE=*keyword*

Specifies the mapping for characters received from the server before they become output. See SET TRANSLATION Keywords for the keywords and their meaning.

The default is /RECEIVE=NONE.

/SEND=*keyword*

Specifies the mapping for characters entered at the keyboard before Client-TELNET sends them to the server. See Table 12-10 for the keywords and their meaning.

The default is /SEND=CR.

Table 12-10 SET TRANSLATION Keywords

Keyword	**Translation**
CR	Client-TELNET translates the carriage return character to a CR/LF sequence
LF	Client-TELNET translates the line feed character to a CR/LF sequence
NONE	Client-TELNET does not translate characters to the CR/LF sequence

SET *[*NO*]*XDISPLOC

Enables or disables setting your current X display location on the remote end, when communicating with a remote TELNET server that also supports this option. Client TELNET checks whether the logical DECW$DISPLAY is defined. If it is, and if the remote server asks for the X display location, the X display server address is transmitted to the remote system.

Use SET NOXDISPLOC before making a connection to disable sending the X display location.

Format

SET XDISPLOC
SET NOXDISPLOC

Example

```
TELNET>SET NOXDISPLOC
TELNET>OPEN ALPHA
$ SHOW DISPLAY
Error opening DECW$DISPLAY as input
No such device available
ALPHA>
```

SHOW OPTIONS

Displays information about the options in effect.

Options modify the way TELNET handles your terminal over the network. When you first establish a connection, both hosts negotiate for the options to use based on the options that each host supports. You can also use the SEND command to change options.

Format

SHOW OPTIONS

Example

```
TELNET>SHOW OPTIONS

Current TELNET options status:

   Remote ECHO
   No remote TRANSMIT-BINARY (normal ASCII)
   No local TRANSMIT-BINARY (normal ASCII)
   Remote SUPPRESS-GO-AHEADS
   Local SUPPRESS-GO-AHEADS
   No remote END-OF-RECORD
   No local END-OF-RECORD
   Local TERMINAL-TYPE: VT300
   Local FLOW-CONTROL: ON
   Local WINDOW-SIZE: 80x35
   Local X-DISPLAY-LOCATION: 192.168.5.195:0.0
```

SHOW STATUS

Displays information about all open TELNET connections and your current TELNET session.

The screen displays the following information:

- Session number, name and internet address of each remote host if a connection is open. An arrow (-->) indicates the current session.
- The list of supported terminal types if no remote connection is open.
- The terminal type used, if a remote connection is open and Client-TELNET negotiated for the terminal type.
- Whether local flow control is ON or OFF.
- Name of the log file if one is open.
- Name of the host character set.
- Name of the terminal character set.
- The current "abort output" (AO), "are you there" (AYT), backward, break (BRK), "erase character" (EC), "erase line" (EL), escape, forward, flush, "interrupt process" (IP), and "go-ahead" (GA) characters (if defined).

Format

SHOW STATUS

Synonym

STATUS

Example

```
TELNET> SHOW STATUS

Client-TELNET V6.0-0  Copyright (c) Process Software
Connected sessions:
    1. bart.nene.com, telnet (192.168.1.92,23).
 -->2. marge.nene.com, telnet (192.168.1.91,23).
"^\" is the escape (attention) character
Current session is operating in 3270 mode.

Terminal type: IBM-3278-2
Local flow control: ON

Keyboard Map File: TCPWARE:MAP3270.DAT

Host Character Set: CANADIAN
Terminal Character Set: LATIN1

"^C" is the escape (attention) character.
```

SHOW TRANSLATION

Displays the current translation settings made using SET TRANSLATION. Both the received and sent translations appear.

Format

SHOW TRANSLATION

Example

```
TELNET> SHOW TRANSLATION
No characters are translated to CRLF when received.
CR is translated to CRLF when sent.
```

SPAWN

Executes DCL commands.

Note! You cannot SPAWN with CAPTIVE accounts.

Format

SPAWN [command-line]

Synonym

Z*[command-line]*

Parameter

command-line

DCL command line that you want executed. If omitted, Client-TELNET spawns an interactive subprocess. To return to TELNET from an interactive subprocess, logout of that subprocess.

Examples

1 This example displays the time on your local host without leaving the TELNET utility:

```
TELNET>SPAWN SHOW TIME
  3-Nov-2014 14:02:48
```

2 This example initiates DCL command mode and returns the DCL prompt:

```
TELNET>SPAWN
$ SHOW TIME
  3-Nov-2014 14:02:51
$ LOGOUT
  Process SMITH_1 logged out at 3-Nov-2014 14:02:54.34
TELNET>
```

To exit the DCL command mode and return to TELNET, enter the LOGOUT command at the DCL prompt.

Chapter 13 TFTP: Trivial File Transfers

Introduction

The Trivial File Transfer (TFTP) utility provides the user interface to TFTP. This program allows a user to transfer files to and from a remote host. TFTP primarily allows remote diskless systems to read bootstrap images over the network. TFTP uses UDP to make transfers. It does not provide user login validation.

See Chapter 4 of the *Installation and Configuration Guide for information* about configuring the TFTP server.

FTP-OpenVMS is a more complete file transfer facility than TFTP.

See Chapter 3, *FTP: Transferring Files*, for details on FTP-OpenVMS.

Invoking TFTP

To invoke TFTP, enter at the DCL prompt:

```
TFTP [host [port]]
```

If you specify a host name, TFTP uses that host for subsequent file transfers. If you also specify a port number, TFTP uses the specified host and port for subsequent file transfers.

Command Reference

You interact with TFTP by typing commands at the `TFTP>` prompt. Client-TFTP supports the following OpenVMS-style commands:

CONNECT	MODE	REXMT	TIMEOUT	HELP
GET	PUT	STATUS	TRACE	QUIT

TFTP offers 20-line recall on the command level.

CONNECT

Sets the host and, optionally, the port number for subsequent file transfers. Note that TFTP uses UDP and, therefore, does not maintain the connection between transfers.

Format

CONNECT *host [port]*

Synonym

OPEN

Parameters

host

Name of the remote host to which you want to connect. The host must exist on the network.

port

Service name or number of the remote port that you want to connect to. The default port number is 69 for read and write requests. You do not need to specify the port number unless you are connecting to a nonstandard server.

Example

Each of these equivalent commands connects to host SIGMA for a file transfer:

```
tftp>connect sigma
tftp>open sigma
```

GET

Gets a file from the previously specified remote host. TFTP writes the local file as a STREAM_LF formatted file.

Since TFTP does not authenticate the client, the server allows access only to files in the directory and its subdirectories defined by the TCPWARE_TFTP_ROOT logical.

The server converts UNIX filenames with their directories into VMS filenames as in Table 13-1. The directory specification is dir and the filename specification with its extension is *filename.ext*.

Table 13-1 TFTP UNIX-to-VMS Filename Conversions

UNIX Filename...	**Is Converted to VMS Filename...**
dir/filename.ext	[.dir]filename.ext
/dir/filename.ext	[.dir]filename.ext

Format

GET *remote-file [local-file]*

Parameters

remote-file

Input file specification on the remote host.

local-file

Output file specification on the local host. If omitted, Client-TFTP uses the remote-file filename and extension.

Examples

1 This command transfers the US-DOMAIN-INFO.TXT file from the previously specified host:

```
tftp>get us-domain-info.txt
```

2 This command transfers the US-DOMAIN-INFO.TXT file from the previously specified host as file LOCALSTUFF.TXT:

```
tftp> get us-domain-info.txt localstuff.txt
```

HELP

Displays a brief help message summarizing the commands.

Format

HELP *[command]*

Parameter

command

Optional command for which you want help.

Examples

This command provides help for the CONNECT and GET commands:

```
tftp>help connect
connect to remote tftpd
tftp> help get
receive file
```

MODE

Sets the file transfer mode to type; type may be either ASCII or BINARY. The initial type is ASCII.

Format

MODE *type*

Parameter

type

The mode type, either ASCII or BINARY.

Example

This command changes the transfer mode to BINARY `(Mode: octet)`:

```
tftp>mode binary
tftp> status
Connected to SIRIUS.nene.com.
Mode:
octet Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
```

PUT

Puts a file to the previously specified remote host.

Since TFTP does not authenticate the client, the server allows access only to files in the directory and its subdirectories defined by the TCPWARE_TFTP_ROOT logical.

The server converts OpenVMS filenames with their directories into UNIX filenames as in Table 13-2. The directory specification is dir and the filename specification with its extension is filename.ext.

Table 13-2 TFTP UNIX-to-VMS Filename Conversions

UNIX Filename...	Is Converted to VMS Filename...
dir/filename.ext	[.dir]filename.ext
/dir/filename.ext	[.dir]filename.ext

Format

PUT *local-file [remote-file]*

Parameters

local-file

Input file specification on the local host.

remote-file

Output file specification on the remote host. If omitted, Client-TFTP uses the *local-file* filename and extension.

Examples

1 This command transfers the US-DOMAIN-INFO.TXT file to the previously specified host:

```
tftp>put us-domain-info.txt
```

2 This command transfers the US-DOMAIN-INFO.TXT file to the previously specified host as file REMOTESTUFF.TXT:

```
tftp>put us-domain-info.txt remotestuff.txt
```

QUIT

Exits the TFTP program. You can also use Ctrl/Z and EXIT to exit the program.

Format

QUIT

Synonyms

EXIT

`Ctrl/Z`

Examples

Each of these equivalent commands exits from TFTP:

```
tftp>quit
tftp>exit
tftp>Ctrl/Z
```

REXMT

Sets the retransmit timer, in seconds. The initial value is 5 seconds.

The value you enter for REXMT is also used together with the specified maximum timeout (set using the TIMEOUT command) to determine the number of times to try and the actual maximum timeout reported in a status request (STATUS).

If the default 5 seconds retransmit interval is used together with the default 25 seconds maximum timeout, the number of times to try is 5, according to the formula:

```
Max-timeout = Rexmt-interval x Tries
```

The REXMT value you enter is always reported (unchanged) on the `Rexmt-interval` line in a STATUS request. However, the maximum timeout may be recalculated before being reported as `Max-timeout`.

See theTIMEOUT command for details on `Max-timeout` recalculation.

Format

REXMT*[time]*

Parameter

time

The time value to set the retransmit timer. If omitted, the value is 5 seconds.

Example

This command changes the retransmit timer (`Rexmt-interval`) to 10 seconds (and the subsequent STATUS command shows the result). The `Max-timeout` is set to five times the `Rexmt-interval` by default.

```
tftp> rexmt 10
tftp>status
Connected to SIRIUS.nene.com.
Mode:
octet Tracing: off
Rexmt-interval: 10 seconds, Max-timeout: 50 seconds
```

STATUS

Displays the current status and parameter settings.

The `Max-timeout` reported is based on the following computation:

```
Max-timeout = Rexmt-interval x Tries
```

The number of tries (`Tries`) is initially 5 unless adjustments are made to the `Max-timeout` and `Rexmt-interval` values (see below for an example).

Note! The total retransmission period (`Max-timeout`) value displayed may be slightly different from that set using the TIMEOUT command. (See theTIMEOUT command for an explanation.)

Format

STATUS

Examples

This command shows the connection status, file transfer mode (`Mode:`), packet trace flag status (`Tracing:`), retransmit timer (`Rexmt-interval:`), and total retransmission period (`Max-timeout:)` values over the period of a number of adjustments. (See theTIMEOUT command for an explanation of the `Max-timeout` recalculations.)

```
tftp> connect spica
tftp>stat
Connected to spica.nene.com.
Mode: netascii Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp>rexmt 4
tftp> stat
Connected to spica.nene.com.
Mode: netascii Tracing: off
Rexmt-interval: 4 seconds, Max-timeout: 20 seconds
tftp>timeout 40
tftp> stat
Connected to spica.nene.com.
Mode: netascii Tracing: on
Rexmt-interval: 4 seconds, Max-timeout: 40 seconds
tftp>timeout 30
tftp>stat
Connected to spica.nene.com.
Mode: netascii Tracing: on
Rexmt-interval: 4 seconds, Max-timeout: 28 seconds
```

TIMEOUT

Sets the total retransmission period, in seconds. The initial value is 25 seconds.

Note! Minor adjustments to the specified retransmission period as reported using STATUS can occur based on concurrent changes made to the retransmit timer setting (REXMT). The retransmission period is calculated based on the following formula:

```
Max-timeout = Rexmt-interval x Tries
```

The Tries value must be an integer value. Thus, if the Max-timeout specified using the TIMEOUT command forms a non-integer ratio with the Rexmt-interval value, the Max-timeout is adjusted accordingly. (See the example.)

Format

TIMEOUT *[time]*

Parameter

time

The total retransmission period, in seconds. If omitted, the value is 25 seconds.

Examples

Note the way in which the retransmission period is adjusted in this example:

```
tftp>connect spica
tftp>stat
Connected to spica.nene.com.
Mode: netascii Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp>rexmt 4
tftp> stat
Connected to spica.nene.com.
Mode: netascii Tracing: off
Rexmt-interval: 4 seconds, Max-timeout: 20 seconds
tftp>timeout 40
tftp>stat
Connected to spica.nene.com.
Mode: netascii Tracing: on
Rexmt-interval: 4 seconds, Max-timeout: 40 seconds
tftp>timeout 30
tftp> stat
Connected to spica.nene.com.
Mode: netascii Tracing: on
Rexmt-interval: 4 seconds, Max-timeout: 28 seconds
```

- The retransmit timer and number of tries are both set to 5 by default, so that initially the `Max-timeout` is 25.
- With the retransmit timer (**`rexmt`**) reset to 4, the `Max-timeout` changes to 4 x 5 = 20.
- Doubling the maximum timeout (**`timeout 40`**), recalculates the number of retries to 40 / 4 = 10.
- Changing the maximum timeout to 30 (with the `rexmt` still set to 4) recalculates the retries to 7, and adjusts the `Max-timeout` to 4 x 7 = 28.

TRACE

Toggles the packet trace flag.

Format

TRACE

Example

This command enables packet tracing. A GET operation shows a timeout on a file transfer read request.

```
tftp>trace
Packet tracing on.
tftp>status
Connected to SIRIUS.nene.com.
Mode: octet Tracing: on
Rexmt-interval: 10 seconds, Max-timeout: 60 seconds
tftp> get rfc999.txt pokertwo.txt
rqst sent RRQ <file=rfc999.txt, mode=octet>
rqst sent RRQ <file=rfc999.txt, mode=octet>
rqst sent RRQ <file=rfc999.txt, mode=octet>
rqst sent RRQ <file=rfc999.txt, mode=octet>
rqst sent RRQ <file=rfc999.txt, mode=octet>
rqst sent RRQ <file=rfc999.txt, mode=octet>
Receive request timed out
```

Chapter 14 Token Authentication: Protecting Logins

Introduction

Token authentication allows you to set additional security restrictions on your FTP, TELNET, RLOGIN, and SET HOST logins. You can set up token authentication through TCPware's Access Control Encryption Client (ACE/Client) on the OpenVMS host, which communicates with Security Dynamics' ACE/Server on a UNIX or Windows NT host. The authentication takes place through a physical SecurID token "smart card" that you use to provide the ACE/Server with the necessary login information.

This chapter explains the TCPware ACE/Client, its interaction with the ACE/Server, and how to enter login information using the SecurID token.

What Is the ACE/Client?

Passwords have long been the front line of defense in protecting hosts and networks, and have come under scrutiny because of well-publicized security breaches. Applications that require passwords to access resources are especially vulnerable to these security breaches.

TCPware's token authentication, in collaboration with Security Dynamics Corporation's Access Control Encryption Server (ACE/Server), works with a two-factor password system to help solve this security problem. Token authentication combines use of the regular login password with a time-based code derived from a token. The authentication system consists of a secure server and the client connected to the devices that need to be protected.

Security Dynamics provides the ACE/Server and a backup server (Slave ACE/Server). TCPware provides the ACE/Client. The ACE/Client handles the interaction between the client and the ACE/Server software at the place where the client is responsible for gathering the authentication data from the user.

The authentication "token" in this case is the Security Dynamics SecurID "smart card," a physical card containing a microprocessor that generates a new, unpredictable code every 60 seconds on its liquid crystal display (LCD). The Server synchronizes and checks this code, when entered, with the user's memorized personal identification number (PIN). These two codes together form the user's PASSCODE.

Token authentication is available for FTP-OpenVMS, TELNET-OpenVMS, RLOGIN, and the OpenVMS SET HOST command.

The TCPware ACE/Client supports Security Dynamics' proprietary encryption (SDI Encryption). The ACE/Server must also use SDI Encryption. The ACE/Server runs on a UNIX or Windows NT machine. The ACE/Client must be registered with the ACE/Server.

Terms

Special terms used in this chapter include:

PIN	Your personal identification number. The PIN consists of four to eight alphanumeric characters. Depending on the policy set by your system manager, either you create your PIN or your system manager creates your PIN.
Duress PIN	Special PIN to use if you are being compromised during the login process.
PASSCODE	Combination of your PIN and the tokencode. If you have a key fob or a standard card, you enter the full PASSCODE (your PIN immediately followed by the current tokencode without a separating space) at the login password prompt. If you have a PINPAD] card, you enter the PIN into your card and then enter the PASSCODE given on the card at the login password prompt.
Tokencode	Random number currently displayed on your Security Dynamics SecurID smart card.

Identifying the SecurID Token Type

SecurID tokens are small, hand-held devices containing microprocessors that calculate and display unpredictable codes. The codes change at a specified interval, typically every 60 seconds.

As an authorized user on a protected system, you are assigned a SecurID token to use when accessing a protected resource. The code displayed on the token at the moment you attempt access is one part of the user's SecurID PASSCODE, which is required for positive authentication and system access. The other part is your valid, memorized PIN.

There are currently three hardware types of SecurID tokens:

Standard SecurID Card	a rectangular card with the tokencode displayed at the upper right hand corner of the card.
SecurID Key Fob	an oblong key fob with a key holder with the tokencode displayed on the center of the fob.
SecurID PINPAD card	a rectangular card with the tokencode displayed at the upper right hand corner and a digit keypad at the bottom from which to enter the PIN.

See theLogging In with a S section.

Login Interfaces

The user interface to token authentication is through login screens for FTP, TELNET, RLOGIN, and SET HOST that display the usual username prompt followed by:

For	**the usual password prompt at which to enter...**
FTP	the PASSCODE.
TELNET, RLOGIN, and SET HOST	your usual password, along with an **`Enter PASSCODE:`** prompt at which to enter the PASSCODE.

Note! For an FTP login, the token cannot be in Next Tokencode or New PIN mode.

Example 14-1 shows a sample FTP login sequence to host BART. The shaded areas show values entered but not displayed on the screen. The PASSCODE is a combination of the PIN and the tokencode when used with a Standard Card or Key Fob.

Example 14-1 FTP Login Sequence Using Token Authentication

```
$ FTP BART
220 bart.process.com (192.168.34.56) FTP-OpenVMS FTPD V6.0-0 (c) Process Software
331 Password required.
230 User logged in, proceed.

_Username [MARGE]: MARGE
331 Password required.
_Password: 192837465
230 User logged in, proceed.
214 SITE +VMS+ recognized.
```

Example 14-2 shows a sample TELNET login sequence to host BART. The shaded areas show values entered but not displayed on the screen. The PASSCODE is a combination of the PIN and the tokencode when used with a Standard Card or Key Fob.

Example 14-2 TELNET Login Sequence Using Token Authentication

```
$ TELNET BART
%TCPWARE_TELNET-I-TRYING, trying BART.nene.com,telnet
(192.168.142.1,23) ...
%TCPWARE_TELNET-I-ESCCHR, escape (attention) character is "^\"

 Welcome to OpenVMS Alpha (TM) Operating System, Version V6.2

Username:   MARGE
Password:   MYPASSWORD

Enter PASSCODE:   192837465
PASSCODE Accepted
```

```
(Bart) $
```

Logging In with a SecurID Token

You may have been assigned one of the following SecurID tokens:

Standard SecurID Card	a rectangular card with the tokencode displayed at the upper right hand corner of the card.
SecurID Key Fob	an oblong key fob with a key holder with the tokencode displayed on the center of the fob.
SecurID PINPAD card	a rectangular card with the tokencode displayed at the upper right hand corner and a digit keypad at the bottom from which to enter the PIN.

To access the protected system, you must enter a valid SecurID PASSCODE], which is made up of two factors:

- Your secret, memorized personal identification number (PIN)
- The tokencode currently displaying on your token

With a conventional security system, it is easy for someone to learn your password and log in under your identity. Requiring two factors ensures reliable identification and authentication.

User Responsibilities

Because this system creates an audit trail that cannot be repudiated, you may be held accountable for activities recorded identifying you as the user. Avoid the unauthorized use of your identity and privileges by protecting the secrecy of your PIN and the possession of your token.

You are responsible for protecting the authentication factors entrusted to you. Keep your PIN secret and protect your SecurID token against loss and theft.

If an unauthorized person learns your PIN and obtains your token, this person can assume your identity. Any action taken by this intruder will be attributed to you in the system's security log.

For your own protection and that of the system, always take the following precautions:

- Never reveal your PIN to anyone. Do not write it down.
- If you think someone learned your PIN, notify the security administrator, who will clear the PIN immediately. At your next login you will have to receive or create a new PIN.
- Exercise care not to lose your SecurID token or to allow it to be stolen. If your token is missing, tell an administrator immediately. The administrator will disable it so that it is useless to unauthorized users.
- Do not let anyone access the system under your identity—do not let them log in with your PIN and a code from your SecurID token.
- It is essential to site security that you follow your system's standard logoff procedures. Failure to log off properly can create a route into the system that is completely unprotected.
- Protect your SecurID token from physical abuse. Do not immerse it in liquids, do not expose it to extreme temperatures and do not put it under pressure or bend it. Each SecurID token comes with care instructions that you should read and follow.

Before You Begin

Have your ACE/Server security administrator fill in the following information before you attempt to log in for the first time:

The system will assign a PIN to you; you cannot create your own
(See theReceiving a S section)

You can use a PIN that you make up yourself
(see theCreating Your Own P section)

Your PIN can contain letters as well as digits
(Applies to the Standard Card and Key Fob only)

All PINs on the system must be the same number of characters: ____
(Applies to the Standard Card and Key Fob only)

All PINs on the system must be the same number of digits: ____
(Applies to the PINPAD card only)

Your PIN can contain from ____ through ____ characters
(Applies to the Standard Card and Key Fob only)

Your PIN can contain from ____ through ____ digits
(Applies to the PINPAD card only)

You can use a duress PIN
(See theUsing a D section)

Receiving a System-Generated PIN

The following steps allow you to use a system-generated PIN:

1 **For PINPAD only:** Clear PIN entries from your card. Press any number on the card, and then press the `P` on the lower right of the card. The display clears and a new tokencode shows after the last of the countdown indicators disappears from the left of the LCD.

Note! For FTP logins, you must first log in on a terminal session such as TELNET or SET HOST to receive your PIN before you can initiate an FTP session.

2 Initiate a terminal login session. After you respond to the usual prompt for your login name, the system asks you to enter a PASSCODE.

3 If you never received a PIN before, enter the code that is currently displaying on your SecurID token at the `Enter PASSCODE` prompt.

If your token previously had a PIN and the administrator did not clear it when setting it in New PIN mode:

- **For Standard Card and Key Fob only:** Enter the old PIN and *right after it*, the code that is currently displaying on your token. (Do not separate the two with a space.)
- **For PINPAD only:** Enter the old PIN into the card and press the diamond (`u`) near the bottom of the card. Then at the `Enter PASSCODE` prompt, enter the code displayed on the card.

4 Press `Return`. If you entered the code incorrectly, the system displays an `Access denied` message. Try again. Once you enter a valid tokencode, the following message appears:

```
Press <Return> to generate a new PIN and display it on screen
                        or
<Ctrl d> to cancel the New PIN procedure:
```

5 If anyone else can see your screen, press `Ctrl/D` so that your secret PIN is not displayed on your screen. The operation is canceled and your card or key fob is still in New PIN mode.

If no one else can see your screen, press **`Return`** to receive your new PIN. Your PIN is displayed for 10 seconds or until you press **`Return`**.

6 Memorize your new PIN. Do not write it down.

7 You are now ready to log in. Wait for the next tokencode, and then follow the instructions in the *Login Steps* section.

Creating Your Own PIN

The following steps allow you to create your own PIN:

1 If you are going to create your own PIN, first give some thought to what it will be. Do not pick an obvious number like a birthday or phone number. See your checklist. You may be allowed letters or digits, or just digits, and the length may be fixed somewhere between four and eight characters, or you may be allowed any number of characters in that range. **For PINPAD only:** PINs cannot begin with a zero.

2 **For PINPAD only:** Clear PIN entries from your card. Press any number on the card, and then press the **`P`** on the lower right of the card. The display clears and a new tokencode shows after the last of the countdown indicators disappears from the left of the LCD.

Note! For FTP logins, you must first log in on a terminal session such as TELNET or SET HOST to receive your PIN before you can initiate an FTP session.

3 Initiate a terminal login session. After you respond to the usual prompt for your login name, the system asks you to enter a PASSCODE.

4 If you never received a PIN before, enter the code that is currently displaying on your SecurID token at the **`Enter PASSCODE`** prompt.

If your token previously had a PIN and the administrator did not clear it when setting it in New PIN mode:

- **For Standard Card and Key Fob only:** Enter the old PIN and *right after it*, the code that is currently displaying on your token. (Do not separate the two with a space.)
- **For PINPAD only:** Enter the old PIN into the card and press the diamond (**`u`**) near the bottom of the card. Then at the **`Enter PASSCODE`** prompt, enter the code displayed on the card.

5 Press **`Return`**. If you entered the code incorrectly, the system displays an **`Access denied`** message. Try again. Once you enter a valid tokencode, you are prompted to perform the New PIN operation.

6 If the prompt reads:

```
Enter your new PIN, containing 4 to 8 characters, or
      Press <Return> to generate a new PIN and display it on screen
        or <Ctrl d> to cancel the New PIN procedure:
```

do one the following and go to Step 8. Otherwise, go to Step 7 now.

- If anyone else can see your screen, press **`Ctrl/D`** to cancel the operation and leave your token in New PIN mode.
- If you want the system to generate a PIN for you and no one else can see your screen, press **`Return`**. Your PIN is displayed for 10 seconds or until you press **`Return`**.
- If you want to create your own PIN and no one else can see your screen, enter the PIN you would like to use, again remembering the guidelines in step 1.

7 If the prompt reads:

```
Enter your new PIN, containing 4 to 8 characters,
or Ctrl/D to cancel the New PIN procedure:
```

then you have to create your own PIN. You cannot have the system generate one for you. If anyone else can see your screen, press `Ctrl/D` to cancel the operation and leave your token in New PIN mode. Otherwise, type in the PIN you would like to use, again remembering the guidelines in Step 1.

8 Memorize your new PIN. *Do not write it down*.

9 You are now ready to log in. Wait for the next tokencode, then follow the instructions in the following *Login Steps* section.

Login Steps

Use the following two steps to log in:

1 Initiate a login session. After you respond to the usual prompt for your login name, you may get your usual password prompt:

- If you are using TELNET, RLOGIN, or SET HOST, enter your usual password at the password prompt and press `Return`. Then go to Step 2.
- If you are using FTP, the password prompt is your PASSCODE prompt. Enter your PIN immediately followed by the code currently displaying on your token, *without any separating space* and press `Return`.

2 At the `Enter PASSCODE:` prompt, enter your PIN immediately followed by the code currently displaying on your token, *without any separating space*.

If you entered a valid PASSCODE, the system displays the message `PASSCODE accepted`.

Once accepted, a SecurID PASSCODE cannot be used again. To log in again, you must wait for a new tokencode to appear. The stack of countdown indicators on the left side of the LCD lets you know how soon the code will be changing.

If the system displays the message `Access denied` instead, you may have typed in your PASSCODE incorrectly. Try again. If you are repeatedly denied access even though you are typing your PASSCODE correctly, contact your system administrator.

"Next Code" Prompt

On the third attempt to log in with a valid PIN but with an invalid tokencode, the system asks you to enter the next code that appears:

```
Please enter the next code from your token:
```

Wait until the stack of countdown indicators on the left side of the LCD tokencode goes down and the code changes, then go ahead and carefully type the new one followed by `Return`.

If you are not granted access after correctly entering the next code, contact your system administrator.

Using a Duress PIN

If your system has the duress PIN option installed, you have two PINs: a regular PIN and a duress PIN. Use your regular PIN for normal logins. Use the duress PIN if you are ever forced to log in by an unauthorized person attempting to gain system access.

If you use your duress PIN, you are granted access and you will see no difference in operation. However, the system notifies administrators that you were forced by an intruder to log in.

Your duress PIN is your regular PIN with 1 added to it but with no carrying. See Table 14-1 for examples.

Table 14-1 Sample Duress PINs

If your regular PIN is...	Then your duress PIN is...	Applies to...
243890	243891	All tokens
243899	243890	All tokens
ABCDEF	ABCDEG	Standard Card and Key Fob
ABCDEZ	ABCDEA	Standard Card and Key Fob

Chapter 15 WHOIS: Username Directory Services

The WHOIS utility allows Internet users to query the Network Information Center (NIC) username directory services.

To invoke WHOIS, enter at the DCL prompt:

```
WHOIS name
```

name is the user's name or other search keyword.

The utility tries to connect to the NIC WHOIS server (`ds.internic.net`) and displays any returned information.

The source code for this utility is in the TCPWARE_COMMON:[TCPWARE.EXAMPLES]WHOIS.C file.

Chapter 16 Accessing Remote Systems with the Secure Shell (SSH) Utilities

The SSH implementation for TCPware provides the client software for allowing secure interactive connections to other computers in the manner of rlogin/rshell/telnet.

The following topics describe how to configure, maintain, and use the following TCPware client and utilities:

- Secure Shell Client (remote login program)
- SSHKEYGEN
- SSHAGENT (authentication agent)
- SSHADD
- CERTTOOL
- CERTVIEW
- CMPCLIENT
- Public-key Subsystem

SSH Protocol Support

The SSH client software supports both the SSH1 and SSH2 protocols. SSH1 and SSH2 are different, and incompatible protocols. The SSH1 implementation is based on the V1.5 protocol and 1.3.7 F-Secure code base, and the SSH2 implementation is based on the V2 protocol and the F-Secure 3.2.0 code base. While SSH2 is generally regarded to be more secure than SSH1, both protocols are offered by TCPware, and although they are incompatible, they may exist simultaneously on server systems, including TCPware servers. The SSH client identifies the protocol(s) offered b3y any given server. If both SSH2 and SSH1 protocols are offered, the client will always use SSH2. Otherwise, the client will use the correct protocol based on the server's capability.

The cryptographic library used by TCPware SSH2 (***this <u>does not</u> apply to SSH1 sessions***) is compiled from unaltered cryptographic source code from F-Secure which is FIPS 140-2 level 2 compliant, as determined by the Computer Security Division of the National Institute of Science and Technology (NIST).

SSH Client Return Status Codes

In versions of TCPware prior to V5.x, the return status codes from the SSH clients listed above were based on UNIX-style status codes, causing problems for many VMS users. Beginning with TCPware V5.x, a logical name may be defined that will cause the SSH clients listed above to use VMS-style return codes. If the logical name isn't defined, the old-style codes will still be used by default. Refer to table C-1 in the *TCPware for OpenVMS User's Guide* for a description of the new status codes.

To enable the new status codes instead of using the pre-TCPware V5.x codes, the logical name TCPWARE_SSH_NEW_STATUS_CODES must be defined system-wide

Secure Shell Client (remote login program)

```
$ SSH hostname[#port] [qualifiers] [command]
```

or

```
$ SSH "user@hostname[#port]" [qualifiers] [command]
```

SSH (Secure Shell) is a program for logging into and executing commands on a remote system. It replaces rlogin, rsh, and telnet, 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.

Table 16-1 SSH Client Command Options and Qualifiers

Qualifier	**Description**
/ALLOW_REMOTE_CONNECT	Allow remote hosts to connect local port forwarding ports. The default is only localhost; may connect to locally binded ports.
/CIPHER=(*cipher-1,...,cipher-n*)	Select encryption algorithm(s).
/COMPRESS	Enable compression.
/CONFIG_FILE=*file*	Read an alternative client configuration file.
/DEBUG=*level*	Set debug level.
/ESCAPE_CHARACTER=*char*	Set escape character; “none” = disable (default: ~).
/HELP	Display help text.
/IDENTITY_FILE=*file*	Identity file for public key authentication.
/IDKEY=(*key1,key2,...,keyn*)	Specifies the key(s) to be used for publickey authentication. If specified, the IDENTIFICATION file is ignored.
/IPV4	Use IPV4 protocol to connect.
/IPV6	Use IPV6 protocol to connect.

/LOCAL_FORWARD=([protocol/]*listen-port:host:port,...)*	Causes the given port on the local (client) host to be forwarded to the given host and port on the remote side. The system to which SSH connects acts as the intermediary between the two endpoint systems. Port forwardings can be specified in the configuration file. Only system can forward privileged ports. See the *Port Forwarding* section for more details.
/LOG_FILE=*logfilename*	Log all terminal activity to the specified log file. Defaults to SYS$DISK:[]SSH.LOG if "*logfilename*" is not specified.
/MAC=*(mac-1,...,mac-n)*	Select MAC algorithm(s).
/NO_AGENT_FORWARDING	Disable authentication agent forwarding.
/NO_X11_FORWARDING	Disable X11 connection forwarding.
/OPTION=*(option-1,...option-n)*	Gives options in the format used in the configuration file. This is useful for specifying options for which there is no separate command-line flag. The option has the same format as a line in the configuration file, and are processed prior to any keywords in the configuration file. For example: /OPTION=(CompressionLevel=6)
/PORT=*port*	Connect to this port on server system. Server must be listening on the same port.
/QUIET	Quiet Mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors display.
/REMOTE_FORWARD=([protocol/]*listen-port:host:port,...*)	Forward remote port to local address. These cause ssh to listen for connections on a port, and forward them to the other side by connecting to host port.
USE_NONPRIV_PORT	Use a non-privileged (>1023) source port.
USER=*user*	Log in to the server system using this user name.
VERBOSE	Display verbose debugging messages. Equal to "/DEBUG=2".
/VERSION	Display version number of the client.

Initial Server System Authentication

When an initial connection is made from the client system to the server system, a preliminary authentication of the server is made by the client. To accomplish this, the server system sends its public key to the client system.

SSH maintains a directory containing the public keys for all hosts to which it has successfully connected. For each user, this is the [.SSH2.HOSTKEYS] directory off the individual SYS$LOGIN directory[1]. In addition, a system-wide directory of known public keys exists in the system directory pointed to by the logical name TCPWARE_SSH2_HOSTKEY_DIR, and this may be populated by the system manager. Both directories are searched as needed when establishing a connection between systems. Any new host public keys are added to the user's HOSTKEYS directory. If a host's identification changes, SSH warns about this and disables password authentication to prevent a trojan horse from getting the user's password. Another purpose of this mechanism is to prevent man-in-the-middle attacks that could be used to circumvent the encryption. The SSH configuration option *StrictHostKeyChecking* can be used to prevent logins to a system whose host key is not known or has changed.

Hostbased Authentication

Hostbased authentication relies on two things: the existence of the user's system and username in either SSH_DIR:HOSTS.EQUIV or in the individual user's SYS$LOGIN:.RHOSTS or SYS$LOGIN:.SHOSTS file; and the server system having prior knowledge of the client system's public host key.

- For SSH2

  When a user logs in:

  1 The server checks the SSH_DIR:HOSTS.EQUIV file, and the user's SYS$LOGIN:.RHOSTS and SYS$LOGIN:.SHOSTS files for a match for both the system and username. Wildcards are not permitted.

  2 The server checks to see if it knows of the client's public host key (SSH2_DIR:HOSTKEY.PUB on VMS client systems) in either the user’s SYS$LOGIN:[SSH2.KNOWNHOSTS] directory or in the system-wide directory pointed to by the TCPWARE_SSH2_KNOWNHOSTS_DIR logical name. The key file is named <FQDN>_<algorithm>.PUB. For example, if the client system is “foo.bar.com” and its key uses the DSS algorithm, the file that would contain its key on the server would be “FOO_BAR_COM_SSH-DSS.PUB”. This key file must exist on the server system before attempting hostbased authentication.

  3 If the key file is found by the server, the client sends its digitally-signed public host key to the server. The server will check the signature for validity.

- For SSH1

  This form of authentication alone is not allowed by the server because it is not secure. The second (and primary) authentication method is the RHOSTS or HOSTS.EQUIV method combined with RSA-based host authentication. It means that if the login would be permitted by .RHOSTS, .SHOSTS, SSH_DIR:HOSTS.EQUIV, or SSH_DIR:SHOSTS.EQUIV file, and if the client’s host key can be verified (see SYS$LOGIN:[.SSH]KNOWN_HOSTS and SSH_DIR:SSH_KNOWN_HOSTS), only then is login permitted. This authentication method closes security holes due to IP spoofing, DNS spoofing, and routing spoofing.

Note! To the administrator: SSH_DIR:HOSTS.EQUIV,.RHOSTS, and the rlogin/rshell protocol are inherently insecure and should be disabled if security is desired.

Publickey Authentication

The SSH client supports DSA-based authentication for SSH2 sessions, and RSA-based authentication for SSH1 sessions. The scheme is based on public-key cryptography. There are cryptosystems where encryption and

[1] In this chapter, the [.SSH] subdirectory in the user’s login directory displays as SYS$LOGIN:[.SSH][.SSH2] displays as SYS$LOGIN:[.SSH2]

decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key.

- For SSH1

  SSH supports RSA-based authentication. The scheme is based on public-key cryptography. There are cryptosystems where encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key.

  RSA is one such system. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key (SYS$LOGIN:[.SSH]AUTHORIZED_KEYS lists the public keys permitted for log in), and only the user knows the private key.

  When the user logs in:

  1 The SSH client program tells the server the key pair it would like to use for authentication.

  2 The server checks if this key pair is permitted.

  If it is permitted, the server sends the SSH client program running on behalf of the user a challenge (a random number) encrypted by the user's public key. The challenge can only be decrypted using the proper private key.

  3 The user's client then decrypts the challenge using the private key, proving that he/she knows the private key but without disclosing it to the server.

  4 SSH implements the RSA authentication protocol automatically.

  The Key Identity files are created with SSHKEYGEN. To create the RSA key pair files with TCPware:

  - Run SSHKEYGEN to create the RSA key pair: IDENTITY and IDENTITY.PUB. Both of these files are stored in the user's SYS$LOGIN:[.SSH]directory. IDENTITY.; is the private key; IDENTITY.PUB is the public key.

  Once you have created your identity files:

  1 Transfer the IDENTITY.PUB file to the remote machine.

  2 Update the AUTHORIZED_KEYS file on the remote machine by appending the contents of the public key file to the SYS$LOGIN:[.SSH]AUTHORIZED_KEYS file on the remote host. The format of the AUTHORIZED_KEYS file requires that each entry consists of a single long line.

  After this, the user can log in without giving the password. RSA authentication is much more secure than rhosts authentication. The most convenient way to use RSA authentication may be with an authentication agent. SeePub for more information.

- For SSH2

  When the user logs in:

  1 The client reads possible keys to be used for authentication from its IDENTIFICATION file. Note that this file does not contain the actual keys; rather, it contains the name of the key files.

  2 The client sends to the server its list of keys.

  3 The server compares each key that it received to see if it can match this key with one of those specified in the AUTHORIZATION file.

  4 The server tells the client the key that was accepted. The client then "signs" the key with a digital signature that only the server with the proper key could verify, and sends the signature to the server.

  5 The server verifies the signature.

Password Authentication

The password is sent to the remote host for checking. The password cannot be seen on the network because all communications are encrypted. When the server accepts the user's identity it either executes the given command or logs into the system and gives the user a normal shell on the remote system. All communication with the remote command or shell will be encrypted automatically.

Using Publickey Authentication with SSH

When a parameter such as a username or hostname is quoted, it's always passed verbatim to the other side. When it's not quoted, it's lowercased. The username entered is used when constructing the digital signature for a key.

On the host side, the uppercase username will be used, and on the server side, the lowercased username (the default on the server since VMS isn't case-sensitive) will be used to generate the digital signature of the public key that's being used, as shown in the following examples:

```
$ SSH2 ″XXXXXXX@HOSTNAME″ command
XXXXXXX is the username that was specified in all uppercase letters.
Public key authentication fails.

$ SSH2 "xxxxxxx@HOSTNAME command
xxxxxx is the username that was specified in all lowercase letters.
Public key authentication is successful.
```

Break-in and Intrusion Detection

Care must be exercised when configuring the client to minimize problems due to intrusion records created by OpenVMS security auditing. The SSH user should consult the system manager to determine the authentication methods offered by the SSH server. Examples of such authentication methods include HostBased, PublicKey, and Password. The client should be configured to not attempt any authentication method that is not offered by the server.

If a client attempts authentication methods not offered by the server, the OpenVMS security auditing system may log several intrusion records for each attempt to create a session to that server. The result being that the user could be locked out and prevented from accessing the server system without intervention from the server's system manager.

Session Termination

The user can disconnect with "~.". All forwarded connections can be listed with "~#". All available escapes can be listed with "~?". A single tilde character can be sent as "~~" (or by following the tilde with a character other than those described above). The escape character must always follow a carriage return to be interpreted as special. The escape character "~" can be changed in configuration files or on the command line.

The session terminates when the command or shell on the remote system exits, or when the user logs out of an interactive session, and all X11 and TCP/IP connections have been closed. The exit status of the remote program is returned as the exit status of SSH.

X11 Forwarding

With X11 in use, the connection to the X11 display forwards to the remote side any X11 programs started from the interactive session (or command) through the encrypted channel. Also, the connection to the real X server is made from the local system. The user should not set DECW$DISPLAY manually. Forwarding of X11 connections can be configured on the command line or in configuration files.

The DECW$DISPLAY value set by SSH points to the server system with a display number greater than zero. This is normal and happens because SSH creates a “proxy" X server on the server system for forwarding the connections over the encrypted channel.

SSH sets up “fake” Xauthority data on the OpenVMS server, as OpenVMS does not support Xauthority currently. It generates a random authorization cookie, stores it in Xauthority on the server, and verifies that any forwarded connections carry this cookie and replace it by the real cookie when the connection is opened. The real authentication cookie is never sent to the server system (and no cookies are sent in plain text).

Configuring the SSH Client

The SSH client uses only SSH2 configuration keywords. There are no SSH1-specific configuration keywords for the SSH client.

The SSH client obtains configuration data from the following sources (in this order):

1 Command line options. See SSH Client Command Options and Qualifiers for details.

2 User’s configuration file (SYS$LOGIN [.SSH2]SSH2_CONFIG). See Table 16-2 for details.

3 System-wide configuration file (SSH2_DIR:SSH2_CONFIG). See SSH2_CONFIG File Configuration Keywords for details.

For each parameter, the first obtained value is used. The configuration files contain sections bracketed by “Host" specifications. That section applies only for hosts that match one of the patterns given in the specification. The matched host name is the one given on the command line. Since the first obtained value for each parameter is used, more host-specific declarations should be given near the beginning of the file, and general defaults at the end.

Table 16-2 SSH2_CONFIG File Configuration Keywords

Keyword	Value	Default	Description
AllowedAuthentications	List	All methods except for hostbased	Permitted techniques, listed in desired order of attempt. These can be the following: keyboard-interactive, password, publickey, kerberos-1@ssh.com, kerberos-tgt-1@ssh.com, kerberos-2@ssh.com, kerberos-tgt-2@ssh.com, and hostbased. Each specifies an authentication method. The authentication methods are tried in the order in which they are specified with this configuration parameter.
AuthenticationSuccessMsg	Y/N	Y	Print message on successful authentication

AuthorizationFile	Filename	Authorization	Authorization file for publickey authentication. See below for more information on the contents of this file.
BatchMode	Y/N	N	Don’t prompt for any input during session
Ciphers	Cipher list	None	Supported encryption ciphers
ClearAllForwardings	Y/N	N	Ignore any specified forwardings
Compression	Y/N	N	Enable data compression
DebugLogFile	Filename	None	Specify the file to hold debug information. If used with the QuietMode keyword turned on as well, only the first part of the log information will be written to SYS$ERROR, until the DebugLogFile keyword is parsed. If QuietMode is not used, all debug output will go to both SYS$ERROR and the log file.
DefaultDomain	Domain		Specify domain name
EscapeChar	Character	“~”	Set escape character (^=ctrl key)
ForwardAgent	Y/N	Y	Enable agent forwarding
ForwardX11	Y/N	Y	Enable X11 forwarding
GatewayPorts	Y/N	N	Allow connection to locally-forwarded ports
Host	Pattern		Begin the per-host configuration section for the specified host

HostCA	Certificate	None	Specifies the CA certificate (in binary or PEM (base64) format) to be used when authenticating remote hosts. The certificate received from the host must be issued by the specified CA and must contain a correct alternate name of type DNS (FQDN). If the remote host name is not fully qualified, the domain specified by configuration option *DefaultDomain* is not fully qualified, the domain specified by configuration option *DefaultDomain* is appended to it before comparing it to certificate alternate names. If no CA certificates are specified in the configuration file, the protocol tries to do key exchange with ordinary public keys. Otherwise, certificates are preferred. Multiple CAs are permitted.
HostCANoCRLs	Certificate	sNone	Similar to HostCA, but disables CRL checking for the given ca-certificate.
IdentityFile	Filename	Identification	Name of identification file for publickey authentication
KeepAlive	Y/N	Y	Send keepalives

LdapServers	ServerURL	None	Specified as *ldap://server.domainname: 389* CRLs are automatically retrieved from the CRL distribution point defined in the certificate to be checked if the point exists. Otherwise, the comma-separated server list given by option *LdapServers* is used. If intermediate CA certificates are needed in certificate validity checking, this option must be used or retrieving the certificates will fail.
LocalForward	Port, Socket		Local port forwarding
Macs	Algorithm	None	Select MAC (Message Authentication Code) algorithm
NoDelay	Y/N	N	Disable Nagle (TCP_NODELAY)
NumberOfPasswordPrompts	Number	3	Number of times the user is prompted for a password before the connection is dropped
PasswordPrompt	String	“%U’s password:”	Password prompt. The following substitutions may be made within the prompt string: %U = insert users’s username %H = insert user’s system name
Port	Port	22	Server port number
QuietMode	Y/N	Y	Quiet mode - only fatal errors are displayed

RandomSeedFile	Filename	Random_seed	Random seed file
RekeyIntervalSeconds	Seconds	3600	Number of seconds between doing key exchanges during a session. 0 = disable
RemoteForward	Port, Socket		Remote port forwarding
SendNOOPPackets	Y/N		Send NOOP packets through the connection. Used typically to prevent a firewall from closing an interactive session
StrictHostKeyChecking	Y/N/Ask	Y	Behavior on host key mismatch
TryEmptyPassword	Y/N	N	Attempt an empty password first when doing password authentication. **Note:** Doing so may result in an extra intrusion being logged.
User	Username		Remote username
VerboseMode	Y/N	N	Verbose mode
VerifyHostKeyDNS	Y/N/ASK	N	Determines if the host key fingerprint must be matched in DNS.

Note! Notes Regarding SSH2_CONFIG

The user may specify default configuration options, called “stanzas”, for different destination systems. The format of this within the configuration file is:

```
hostname:
  keyword                      value
  keyword        value

hostname2:
   keyword       value
   keyword       value
```

For example:

```
petunia:
  port          17300
  user          dilbert
  host          petunia.flowers.com

rose:
  port          16003
  user          dogbert
  host          rose.flowers.com
  allowedauthentications password

*.beans.com:
  user           limabean
  keepalive      no
  ciphers        3des,twofish
```

In the preceding example:

- When a user types "$ SSH PETUNIA", the client will connect to port 17300 on petunia.flowers.com, and will use the default username of "dilbert".
- When a user types "$ SSH ROSE", the client will connect to port 16003 on host rose.flowers.com, and will use the default username of "dogbert", and only allow password authentication.
- When a user types "$ SSH <anything>.BEANS.COM", the client will use the default username of "limabean", will not send keepalives, and will only allow 3DES or TWOFISH encryption.

The user may override defaults specified in configurations. Options that are specified on the command line override any like options in the configuration file. For example, if the user wants to use a username of "catbert" when connecting to host rose instead of the default username of "dogbert", this would be specified as:

```
$ SSH /USER=CATBERT ROSE
```

Authorization File Options

The authorization file has the same general syntax as the configuration files. The following keywords may be used.

Key

This is followed by the filename of a public key in the [.SSH2] directory file that is used for identification when contacting the host. If there is more than one key, they are all acceptable for login.

Options

This keyword, if used, must follow the "Key" keyword above. The various options are specified as a comma-separated list. See below for documentation of the options.

Command

This keyword is deprecated (though it still works). Use Options instead.

Options that can be specified:

allow-from and deny-from

Specifies that in addition to public-key authentication, the canonical name of the remote host must match the pattern(s). These parameters follow the logic of {Allow,Deny} Hosts described in detail in sshd2_config. Specify one pattern per keyword, and multiple keywords can be used.

command="command"

This is used to specify a "forced command" that will be executed on the server side instead of anything else when the user is authenticated. This option might be useful for restricting certain public keys to perform just a specific operation. An example might be a key that permits remote backups but nothing else. Notice that the client may specify TCP/IP and/or X11 forwarding, unless they are explicitly prohibited.

idle-timeout=time

Sets idle timeout limit to time in seconds (s or nothing after number), in minutes (m), in hours (h), in days (d), or in weeks (w). If the connections have been idle (all channels) for that long a period of time, the connection is closed down.

no-port-forwarding

Forbids TCP/IP forwarding when this key is used for authentication. Any port forward requests by the client will return an error. This might be used, for example, in connection with the command option.

no-x11-forwarding

Forbids X11 forwarding when this key is used for authentication. An X11 forward request by the client will return an error.

SSH Client/Server Authentication Configuration Examples

Hostbased Authentication Example

The following is an example of how to set up the SSH client and SSH2 server for Hostbased Authentication:

```
$!
$! First, generate the host key - ONLY if it doesn't exist!
$!
$ netcu sshkeygen /ssh2 /host
Generating 1024-bit dsa key pair
4 oOo.oOo.oOo

Key generated.
1024-bit dsa, myname@myclient.foo.com, Thu MAR 04 2014 13:43:54
Private key saved to tcpware_ssh2_hostkey_dir:hostkey.
Public key saved to tcpware_ssh2_hostkey_dir:hostkey.pub

$ directory tcpware_ssh2_hostkey_dir:hostkey.*

Directory TCPWARE_SPECIFIC:[TCPWARE.SSH2.HOSTKEYS]

HOSTKEY.;1          HOSTKEY.PUB;1

Total of 2 files
$!
$! Copy the client system public key to the user directory on the
$! server
$!
$! DECnet must be running before you execute the following
$! commands:
$!
$ copy tcpware_ssh2_hostkey_dir:hostkey.pub -
_$ myserv"myname myuser"::[.ssh2.knownhosts]myclient_foo_com_ssh-dss.pub
$!
$! Finally, log into the server system and ensure the
$! SSH_DIR:HOSTS.EQUIV file is correct
$!
$ SET HOST MYSERV

      Welcome to OpenVMS (TM) VAX Operating System, Version V7.3

Username: myname
Password:
      Welcome to OpenVMS VAX V7.3

    Last interactive login on Monday,  1-MAR-2014 17:07
    Last non-interactive login on Monday, 1-MAR-2014 08:30

MYSERV_$ type tcpware:hosts.equiv
#
# HOSTS.EQUIV - names of hosts to have default "r" utility access
# to the local # system.
#
#   This file should list the full domain-style names.
#
```

```
#   This list augments the users' SYS$LOGIN:.RHOSTS file for
#   authentication.
#   Both the .RHOSTS and the HOSTS.EQUIV files are cached by
#   TCPWARE_ - see the section entitled "RLOGIN and RSHELL
#   Authentication Cache" in the _Administrator's Guide_ for more
#   information on controlling the cache.
#
#   This file is ignored for the users SYSTEM and ROOT. SYSTEM and
#   ROOT must have a SYS$LOGIN:.RHOSTS file if you want to use
#   RSHELL or RLOGIN with them.
#
localhost
myclient.foo.com       myname
MYSERV_$
MYSERV_$ logout
  MYNAME      logged out at 1-MAR-2014 13:46:58.91
%REM-S-END, control returned to node MYCLIENT::
```

Publickey Authentication Example

The following is an example of how to set up the SSH client and SSH2 server for Publickey Authentication:

```
$!
$! First, generate a key tuple
$!
$ netcu sshkeygen /ssh2
Generating 1024-bit dsa key pair
   1 oOo.oOo.oOo.

Key generated.
1024-bit dsa, myname@myclient.foo.com, Thu Mar 04 2014 14:06:10
Passphrase :
Again      :
Private key saved to DISK$USERDISK:[MYNAME.SSH2]id_dsa_1024_a.
Public key saved to DISK$USERDISK:[MYNAME.SSH2]id_dsa_1024_a.pub
$ directory [.ssh2]id*.*/since = TODAY

Directory DKA0:[MYNAME.SSH2]

ID_DSA_1024_A.;1    ID_DSA_1024_A.PUB;1

Total of 2 files.
$!
$! Now create the IDENTIFICATION. file.  This contains the name of
$! all the keys you wish to use for public-key authentication.
$!
$ set default [.ssh2]
$ copy tt: identification.
  idkey id_dsa_1024_a
  ^Z
$!
$! Copy the key to the user's [.ssh2] directory on the server
$! system
$!
$ copy id_dsa_1024_a.pub myserv"myname mypass"::[.ssh2]
$!
$! Now log into the server system and create the AUTHORIZATION
$! file
```

```
$!
$ set host myserv

      Welcome to OpenVMS (TM) VAX Operating System, Version V7.3

Username: myname
Password:
      Welcome to OpenVMS VAX V7.3

    Last interactive login on Tuesday,  2-MAR-2014 13:46
    Last non-interactive login on Tuesday,  2-MAR-2014 13:47

$ set default [.ssh2]
$ directory [.ssh2]id*.*

Directory DKA0:[MYNAME.SSH2]

ID_DSA_1024_A.PUB;1

Total of 1 file.
$ copy tt: authorization.
key id_dsa_1024_a.pub
^Z
$ logout
  MYNAME      logged out at 2-MAR-2014 14:10:26.16
%REM-S-END, control returned to node MYCLIENT::
```

Note! The public key assistant and subsystem can also be used to transfer public keys and maintain the authorization file to implementations that support the public key subsystem.

SSH1 Example

```
$ ! An example of the procedure of setting up SSH to enable
$ ! RSA-based authentication.
$ ! Using SSH client node to connect to an SSH server node.
$ !
$ ! On the client node
$ !
$ NETCU SSHKEYGEN /SSH1
Initializing random number generator...
Generating p:  ..............................++ (distance 662)
Generating q:  ................++ (distance 370)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key
(DISK$SYS_LOGIN:[MYNAME.ssh]identity.):
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in
DISK$SYS_LOGIN:[MYNAME.ssh]identity..
Your public key is:
1024 33 13428..........29361 MYNAME@long.hair.com
Your public key has been saved in DISK$SYS_LOGIN:[MYNAME.ssh]identity.pub
$ !
$ ! A TCP/IP stack must be loaded on the remote system.
$ !
$  FTP DAISY /USER=MYNAME/PASSWORD=DEMONSOFSTUPIDITY -
```

```
_$ PUT DISK$SYS_LOGIN:[MYNAME.ssh]identity.PUB -
_$ DISK$SYS_LOGIN:[MYNAME.ssh]identity.PUB
long.hair.com TCPware FTP user process V6.0(119)
Connection opened (Assuming 8-bit connections)
<daisy.hair.com TCPware FTP Server Process V6.0(16) at Thu 6-Mar-2013 3:20PM-EDT
[Attempting to log in as myname]
<User MYNAME logged into DISK$SYS_LOGIN:[MYNAME] at Thu 6-MAR-2013 3:21PM-EDT, job
20e00297.
<VMS Store of DISK$SYS_LOGIN:[MYNAME.SSH]IDENTITY.PUB; started.
<Transfer completed.  395 (8) bytes transferred.
<QUIT command received. Goodbye.
$
$ TELNET DAISY
Trying... Connected to DAISY.HAIR.COM.


        Authorized Users Only (TM) VAX Operating System, Version V7.1

Username: MYNAME
Password:
     Welcome to OpenVMS (TM) VAX Operating System, Version V7.1 on node DAISY
    Last interactive login on Thursday,  4-MAR-2014 08:07
    Last non-interactive login on Thursday,  6-MAR-2014 15:21
         Logged into DAISY at  4-MAR-2014 15:22:43.68
$ !
$ ! For the first entry into the AUTHORIZED_KEYS file copy
$ ! (or rename) the file [.SSH]IDENTITY.PUB to
$ ! [.SSH]AUTHORIZED_KEYS.
$ !
$ COPY [.SSH]IDENTITY.PUB [.SSH]AUTHORIZED_KEYS.
$
$ ! FOR SUBSEQUENT ENTRIES use the APPEND command
$ !
$ APPEND [.SSH]IDENTITY.PUB [.SSH]AUTHORIZED_KEYS.
$
$ ! A sanity check of the file protections shows
$ !
$ DIRECTORY/PROTECTION [.SSH]*.*

Directory DISK$SYS_LOGIN:[MYNAME.SSH]

AUTHORIZED_KEYS.;1   (RWE,RWED,RE,E)
IDENTITY.;1          (RWD,RWD,,)
IDENTITY.PUB;1       (RWE,RWED,RE,E)
KNOWN_HOSTS.;1       (RWD,RWD,,)
RANDOM_SEED.;1       (RWD,RWD,,)

Total of 5 files.
$ !
$ DIRECTORY/PROTECTION SSH.DIR

Directory DISK$SYS_LOGIN:[MYNAME]

SSH.DIR;1            (RWD,RWD,,)

Total of 1 file.
```

SSH2 User Authentication Using Certificates:

Client setup:

1. Copy the private key and certificate (.crt) into the user’s [.ssh2] directory, and edit the [.ssh2] identification file, adding entry “certkey private key name”.

```
$dir [.ssh2]

Directory DKA0:[DILBERT.SSH2]

AUTHORIZATION.;13 IDENTIFICATION.;1 MYCERT.;1 MYCERT1.CRT.;2

Total of 4 files.

$ type [.ssh2]identification.

  certkey mycert1

  $
```

Server setup:

1. Copy the CA certificate into your SSH2_DIR: directory.
2. Add the following entries in SSH2_DIR:SSHD2_CONFIG:
   ```
   Pki SSH2_DIR:<CAcertname>
   Mapfile SSH2_DIR:<CAcertname>.map
   ```

   The *Pki* keyword begins an authority block for a given CA certificate. There might be more than one CA certificate along with its own mapping file.

   The *Mapfile* keyword specifies the location of the certificate to username mapping file.

   In addition, for testing, you might use PkiDisableCRLs yes to disable CRL checking for the given authorization block.
3. Create the mapping file SSH2_DIR:<CAcertname>.map

   The mapping file consists of rows of the following format:

   ```
   userid mappingrule mapdata
   ```

   Userid is the userid that’s allowed to login for the given cert (there might be multiple userids for a given certificate).

   Mapping rule is one of *subject*, *email*, *serialandissuer* and *emailregex*.

   Subject means that the following mapdata is matched against the subject of the certificate:

   Email is the e-mail alternative subject extension (with emailregex can be used regular expressions - e.g., %subst% emailregex ([a-z}|+)@foo\.com would be any trusted certificate having e-mail alternative name of <username>@foo.com to login with userid <username>)

   SerialAndIssuer is the serial number and DN of the issuer separated by whitespace.

   DNs are used in reverse LDAP order (e.g., c=US,o=Foobar,cn=Dilbert Dogbert).

SSH2 Hostkey Authentication Using Certificates

Server setup:

1. Create a certificate for the server. Host certificate must contain FQDN as DNS alternative name.

2. Copy the private key and certificate into TCPWARE_SSH2_HOSTKEY_DIR directory.

3. Add the following entries into ssh2_dir:sshd2_config file

```
HostKeyFile tcpware_ssh_hostkey_dir:<hostcert>
HostCertificateFile tcpware_ssh_hostkey_dir:<hostcert>.crt
```

Client setup:

1. Copy the CA certificate in TCPWARE_SSH2_HOSTKEY_DIR directory.

2. Add the following entries into ssh2_dir:ssh2_config

```
HostCA tcpware_ssh_hostkey_dir:<CAcert>.crt
DefaultDomain <domain of the FQDN of the client>
```

Note! For testing purposes, you can use HostCANoCRLs instead of HostCA to disable CRL checking.

Host Key Verification Using DNS

TCPware SSH can be configured to calculate the fingerprint of the host key it receives, then perform a lookup in DNS for that fingerprint. This can help prevent man-in-the-middle attacks. See RFC 4255 for more details. Refer to Appendix D of the *TCPware for OpenVMS Management Guide* for information on configuring DNSSEC on TCPware systems.

In order to do this, the following conditions must be met:

- DNSSEC must be enabled and configured for the DNS used by the client.
- A host key SSHFP record must be generated, signed by the zone key, and added to the DNSSEC configuration as a type SSHFP record for the server system. The TCPware SSH-KEYGEN2 utility can be used to display the host key SSHFP type record for DNS.
- The client configuration keyword VerifyHostKeyDNS must be set to “Y” or “ASK”
- The client configuration keyword StrictHostKeyChecking must be set to “Y” or “ASK”.

When the host key is received from the server, and after the client goes through its normal host key checking (e.g., does the client already know about this host key), it checks the status of the VerifyHostKeyDNS keyword. If not set to “N”, the client calculates the fingerprint of the host key, then performs a DNS lookup of the key,

If no records are found, the user may be given the option of proceeding (if VeifyHostKeyDNS is set to “ASK”). If the user responds “N”, then the session is terminated. Otherwise, the host key is accepted and the session continues.

If one or more records are found, the fingerprint and type of the host key received are compared against those found in DNS. If no matches are made, the user may be given the opportunity to ignore this state (see above).

If a match was made, the RRSET_VALIDATED flag returned by DNSSEC is examined to see if the signing of the records can be fully trusted. If this is true, the host key processing is complete. If this flag is false, the user may be given the opportunity to ignore this state (see above).

Port Forwarding

Port forwarding is a mechanism whereby programs that use known TCP/IP ports can have encrypted data forwarded over unsecure connections. This is also known as "tunneling".

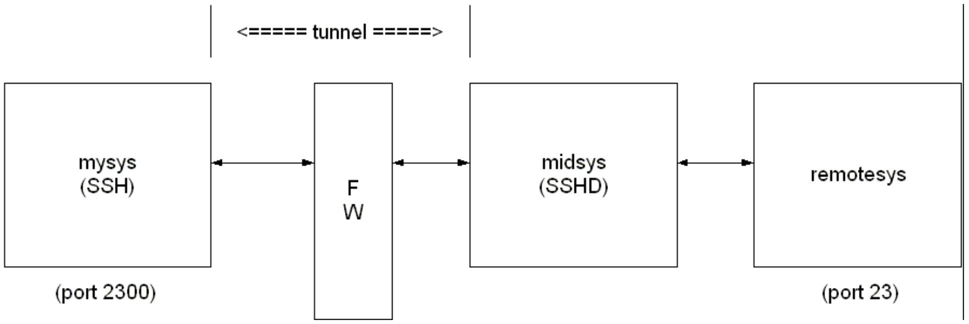
If the user is using an authentication agent, the connection to the agent is forwarded automatically to the remote side unless disabled on the command line or in a configuration file. Forwarding of arbitrary TCP/IP connections over the secure channel can be specified either on the command line or in a configuration file.

Note! Forwarded ports (tunnels) exist only as long as the SSH session that established them exists; if the SSH session goes away, so do the forwardings.

```
/LOCAL_FORWARD=(localport:remotehost:remoteport)
```

This causes `localport` on the system the client is running on to be forwarded to `remotehost:remoteport`. The system to which SSH2 connects acts as the intermediary between the two endpoint systems.

For example: Use port forwarding to allow a system (`midsys)` to encrypt and forward TELNET sessions between itself (`mysys`) that's outside a corporate firewall to a system (`remotesys`) that is inside a corporate firewall. Note that the use of port 2300 in the examples is arbitrary.



From the DCL prompt on `mysys`:

```
$ SSH midsys /local_forward=(2300:remotesys:23)
```

With the SSH session to `midsys` now active, type in another window on `mysys`:

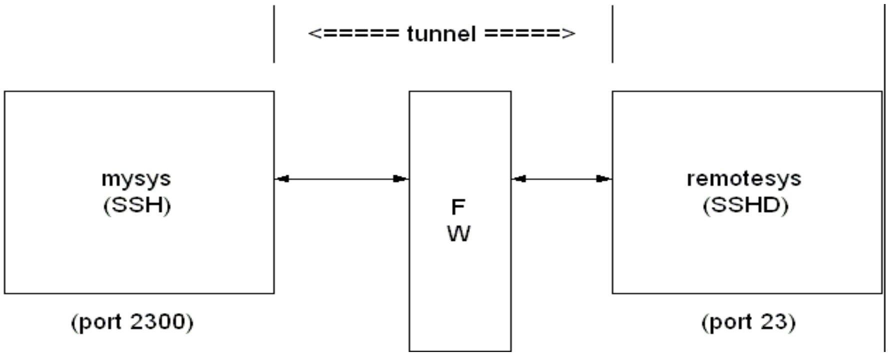
```
$ telnet localhost /port=2300
```

Note! The SSH session must remain active for port forwarding activity.

This causes a connection to `mysys:2300`. The SSH2 client has bound to this port, and will see the connection request. SSH sends an "open channel" request to `midsys`, telling it there's a connect request for port 23 on `remotesys. Midsys` will connect to `remotesys:23`, and send back the port information to `mysys`. `Mysys` completes the connection request, and the TELNET session between `mysys` and `remotesys` is now in place, using the tunnel just created through the firewall between `mysys` and `midsys`.

All traffic between `mysys` and `midsys` (through the firewall) is encrypted/decrypted by SSH on `mysys` and SSHD on `midsys`, and hence, is safe. TELNET does not know this, of course, and does not care.

Note that ports can also be forwarded from a localhost to the remotehost that's running SSHD, as illustrated in this figure.



In this example, port 2300 on `mysys` is being forwarded to `remotesys:23`. To do this, use SSH on `mysys`:

```
$ SSH remotesys /local_forward=(2300:remotesys:23)
```

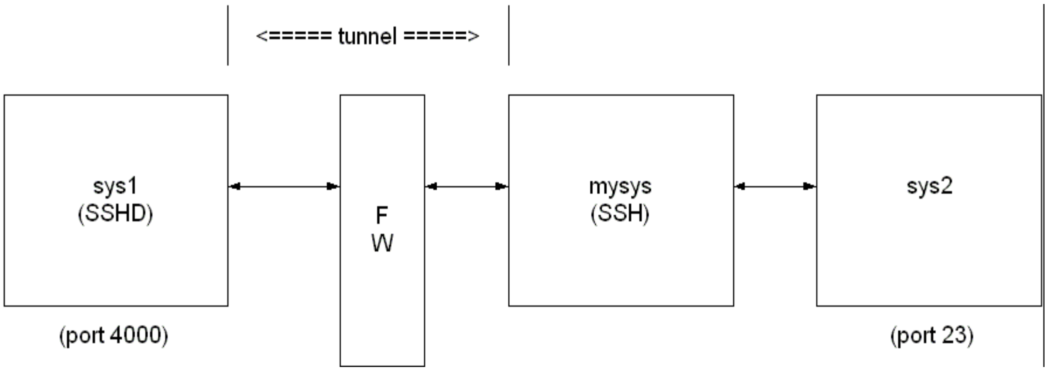
Then, also on `mysys`, type:

```
$ telnet localhost /port=2300
```

When SSH and SSHD start their dialog, SSHD on `remotesys` connects back to itself, port 23, and the TELNET session is established.

```
/REMOTE_FORWARD=(remoteport1:remotehost:remoteport2)
```

This causes `remoteport1` on the system to which SSH connects to be forwarded to `remotehost:remoteport2`. In this case, the system on which the client is running becomes the intermediary between the other two systems.



For example, a user wants to use `mysys` to create a tunnel between `sys1:4000` and `sys2:23`, so that TELNET sessions that originate on `sys1:4000` get tunneled to `sys2` through the firewall. On `mysys`:

```
$ SSH sys1 /remote_forward=(4000:sys2:23)
```

Now, on `sys1`, a user could establish a TELNET session to `sys1` by doing:

```
$ telnet localhost /port=4000
```

The mechanism used for making the TELNET connection (setting up the tunnel) is essentially the same as described in the /LOCAL_FORWARD example above, except that the roles of SSH and SSHD in the dialog are reversed.

Other Files

The files in Table 16-3 are used by SSH. Note that these files generally reside in the [.SSH2] subdirectory from the user's SYS$LOGIN directory. The [.SSH2] subdirectory is created automatically on your local system the first time SSH is executed, and on a remote OpenVMS system the first time an SSH connection is made to that system. File protection for SYS$LOGIN:SSH2.DIR should be (S:RWD, O:RWD, G:, W:).

Table 16-3 SSH2 Files

File Name	**Resides On**	**Description**
[.SSH2]SSH2_CONFIG.	Client System	This is the individual configuration file. This file is used by the SSH2 client. It does not contain sensitive information. The recommended file protection is (S:RWD,O:RWD,G:,W:).
[.SSH2]IDENTIFICATION	Client System	Contains the information about private keys that can be used for public-key authentication, when logging in.
[.SSH2]ID_*alg_bits_seq*	Client System	Contains a private key for authentication. • *alg* is either RSA or DSA • *bits* is the length of the key • *seq* is an incrementing alphabetic value Thus, a key named `ID_DSA_1024_A`. indicates this is a private DSA key 1024 bits long, and it is the first time the key was generated using SSHKEYGEN. A user may have multiple private key files in a directory.

[.SSH2]ID_*alg_bits_seq*.PUB	Client System and Server System	Contains a public key for authentication. • *alg* is either RSA or DSA • *bits* is the length of the key • *seq* is an incrementing alphabetic value Thus, a key named ID_DSA_1024_B.PUB indicates this is a public DSA key 1024 bits long, and it is the second time the key was generated using SSHKEYGEN. A user may have multiple public key files in a directory.
[.SSH2.HOSTKEYS]xxx.PUB	Client System	Contains public host keys for all hosts the user has logged into. The files specifications have the format KEY_*port_hostname*.PUB • *port* is the port over which the connection was made • *hostname* is the hostname of the key's host. For example, if tulip.flowers.com was accessed via port 22, the keyfile would be "KEY_22_TULIP_FLOWERS_ COM.PUB". If this file changes on the host (for example, the system manager regenerates the host key), SSH2 will note this and ask if you want the new key saved. This helps prevent man-in-the-middle attacks.

[.SSH2]RANDOM_SEED.	Client System	Seeds the random number generator. This file contains sensitive data and MUST have a protection of no more than (S:RWD,O:RWD,G:,W:), and it must be owned by the user. This file is created the first time the program is run and is updated automatically. The user should never need to read or modify this file. On OpenVMS systems, multiple versions of this file will be created; however, all older versions of the file may be safely purged. Use the DCL command: SET FILE /VERSION_LIMIT=n RANDOM_SEED to set a limit on the maximum number of versions of this file that may exist at any given time.
SSH_DIR:.RHOSTS	Server System	Is used in hostbased authentication to list the host/user pairs that are permitted to log in. Each line of the file contains a host name (in the fully-qualified form returned by name servers), and then a user name on that host, separated by a space. This file must be owned by the user, and must not have write permissions for anyone else. The recommended permission is read/write for the user, and not accessible by others.
SSH_DIR:.SHOSTS	Server System	Is used the same way as .RHOSTS.
TCPWARE:HOSTS.EQUIV	Server System	Is used during .rhosts authentication. It contains fully-qualified hosts names, one per line. If the client host is found in this file, login is permitted provided client and server user names are the same. Additionally, successful RSA host authentication is required. This file should only be writable by SYSTEM.

TCPWARE:SHOSTS.EQUIV	Server System	Is processed exactly as `SSH_DIR:HOSTS.EQUIV`. This file may be useful to permit logins using SSH but not using rshell/rlogin.
SSH2_DIR:SSH2_CONFIG	Client System	This is a system-wide client configuration file. This file provides defaults for those values that are not specified in a user's configuration file, and for users who do not have a configuration file. This file must be world-readable.
TCPWARE_SSH2_KNOWNHOSTS_DIR	Server System	Contains public host keys for all hosts the system has logged into. The files specifications have the format KEY_*port*_*hostname*.PUB • *port* is the port over which the connection was made • *hostname* is the hostname of the key's host. For example, if tulip.flowers.com was accessed via port 22, the keyfile would be "KEY_22_TULIP_FLOWERS_ COM.PUB". If this file changes on the host (for example, the system manager regenerates the host key), SSH will note this and ask if you want the new key saved. This helps prevent man-in-the-middle attacks.

SSHKEYGEN

Generates authentication key pairs. The format of the keys is incompatible between SSH1 and SSH2. Therefore, the correct format keys must be generated for each version of the protocol to be supported.

There is no way to recover a lost passphrase. If the passphrase is lost or forgotten, you need to generate a new key and copy the corresponding public key to other systems.

Each key may be protected via a passphrase, or it may be left empty. Good passphrases are 10-30 characters long and are not simple sentences or otherwise easily guessable. Note that the passphrase can be changed later, but a lost passphrase cannot be recovered, as a "one-way" encryption algorithm is used to encrypt the passphrase.

Note! The Host Key has no password.

SSH1

```
NETCU SSHKEYGEN /SSH1 [/BITS=n] [/IDENTITY_FILE=file]
                         [/PASSPHRASE=passphrase] [/COMMENT=comment]
NETCU SSHKEYGEN /SSH1 /CHANGE_PASSPHRASE [/PASSPHRASE=old_passphrase]
                         [/NEW_PASSPHRASE=new_passphrase]
NETCU SSHKEYGEN /SSH1 /CHANGE_COMMENT [/PASSPHRASE=passphrase]
                         [/COMMENT=comment]
NETCU SSHKEYGEN /SSH1 /CHANGE_CIPHER [/IDENTITY_FILE=file]
                         [/PASSPHRASE=passphrase]
NETCU SSHKEYGEN /SSH1 /HOST [/BITS=n][/COMMENT=comment]
```

Table 16-4 SSH1 SSHKEYGEN Options

Option	Description
/BITS=*nnn*	Specify key strength in bits (default = 1024).
/CHANGE_PASSPHRASE	Change the passphrase of private key file.
/CHANGE_COMMENT	Change the comment for a key.
/CHANGE_CIPHER	Change the cipher to current default (3DES).
/COMMENT="*comment*"	Provide the comment.
/HOST	Generate the host key.
/IDENTITY_FILE=*file*	Specify the name of the host key file.
/PASSPHRASE=*ppp*	Provide the current passphrase.

/NEW_PASSPHRASE=*ppp*	Provide new passphrase.
/VERSION	Print sshkeygen version number.

SSH2

```
NETCU SSHKEYGEN /SSH2[/BITS=n][/COMMENT=comment][/KEYTYPE=type]
                     [/KEYS=(key1...keyn)]
                     [/PASSPHRASE=ppp|/NOPASSPHRASE][/STIR=file][/QUIET]
NETCU SSHKEYGEN /SSH2/HOST
                     [/BITS=n][/COMMENT=comment][/STIR=file][/QUIET]
NETCU SSHKEYGEN /SSH2/DERIVE_KEY=file
NETCU SSHKEYGEN /SSH2/EDIT=file
NETCU SSHKEYGEN /SSH2/FINGERPRINT=file
NETCU SSHKEYGEN /SSH2/INFO=file [/BASE=n]
NETCU SSHKEYGEN /SSH2/SSH1_CONVERT=file
NETCU SSHKEYGEN /SSH2/X509_CONVERT=file
NETCU SSHKEYGEN /SSH2/PKCS_CONVERT=file
NETCU SSHKEYGEN /SSH2/EXTRACT_CERTS=file
NETCU SSHKEYGEN /SSH2/HELP
NETCU SSHKEYGEN /SSH2/VERSION
```

NETCU SSHKEYGEN /SSH2/NOWARN
NETCU SSHKEYGEN /SSH2/DNS_DIGEST

Option	**Description**
/BASE=*nnn*	Number base for displaying key info
/BITS=*nnn*	Specify key strength in bits (default = 1024).
/COMMENTS=”*comment*”	Provide the comment.
/PKCS_CONVERT=*file*	Convert a PKCS 12 file to an SSH2 format certificate and private key.
/SSH1_CONVERT=*file*	Convert SSH1 identity to SSH2 format.
/X509_CONVERT=*file*	Convert private key from X.509 format to SSH2 format.
/DERIVE_KEY=*file*	Derive the private key given in ‘file’ to public key.
/DNS_DIGEST	Calculate and display a DNSSEC SSHFP record of the local host key that can be added to a DNS configuration file.

/EDIT=*file*	Edit the comment/passphrase of the key.
/EXTRACT_CERTS=*file*	Extract certificates from a PKCS 7 file.
/FINGERPRINT=*file*	Dump the fingerprint of file.
/INFO=*file*	Load and display information for ‘file’.
/HELP	Print help text.
/HOST	Generate the host key.
/KEYS=(*key1,...,keyn*)	Generate the specified key file(s).
/KEYTYPE=(*dsa* | *rsa*)	Choose the key type: dsa or rsa.
/OPENSSH_CONVERT=*file*	Convert the specified OpenSSH key to SSH2 format
/OUTPUT_FILE=*file*	Write the key to the specified output file
/PASSPHRASE=*ppp*	Provide the current passphrase.
/NOPASSPHRASE	Assume an empty passphrase.
/QUIET	Suppress the progress indicator.
/STIR=file	Stir data from file to random pool.
/VERSION	Print sshkeygen version number.
/[NO]WARN	Enable or disable warnings if the process of generating host keys using /HOST will cause existing host keys to be overwritten. If enabled, the user will be prompted to overwrite them. If disabled, no warnings or prompts are issued if the host keys exist. Default is /WARN.

There is also a comment field in the public key file that is for the convenience to the user to help identify the key. The comment can tell what the key is for, or whatever is useful. The comment is initialized to `nnn-bit dsa, username@hostname, ddd mm-dd-yyyy hh:mm:ss` when the key is created unless the /COMMENT qualifier is used, and may be changed later using the /EDIT qualifier.

Note! When the /HOST qualifier is used, the /KEYS=(key1,...keyn) qualifier is ignored.

Note! The public key file must be world-readable.

SSHAGENT (authentication agent)

```
SSHAGENT
```

DESCRIPTION

SSHAGENT is a program that holds authentication private keys. Both SSH1 and SSH2 keys are supported by SSHAGENT. SSHAGENT may be started in the beginning of a login session by including the commands to start it in, for example, LOGIN.COM. It may also be started interactively at any time during a login session.

To start SSHAGENT, one of the three methods may be used:

1. Start it in a separate window:

```
 $ SSHAGENT
```

2. Spawn it as a subprocess:

```
 $ SPAWN/NOWAIT SSHAGENT
```

3. Run it in a detached process:

```
 $ RUN/DETACHED/OUTPUT=AGENT.OUT/INPUT=NLA0:/PROCESS_NAME="SSH AGENT"
  SSH_EXE:SSH-AGENT2
```

The agent is used for Publickey Authentication when logging to other systems using SSH. A connection to the agent is available to all programs run by all instances of the user on a specific system. The name of the mailbox used for communicating with the agent is stored in the TCPWARE_SSH_AGENT_ *username* logical name. Note that while the agent mailbox is accessible only by the user that starts the agent, a user with sufficient VMS privileges could access the agent mailbox and steal or modify keys currently loaded into the agent (although, the keys as stored on disk cannot be modified simply by accessing the agent).

The agent does not have any private keys initially. Keys are added using SSHADD. When executed without arguments, SSHADD adds the user's identity files. If the identity has a passphrase, SSHADD asks for the passphrase. It then sends the identity to the agent. Several identities can be stored in the agent; the agent can use any of these identities automatically.
$ SSHADD /LIST displays the identities currently held by the agent. The idea is that the agent is run on the user's workstation.

FILES

[.SSH]IDENTITY in SYS$LOGIN:	Contains the RSA authentication identity of the user. This file should not be readable by anyone but the user. It is possible to specify a passphrase when generating the key. That passphrase is used to encrypt the private part of this file. This file is not used by SSHAGENT, but is added to the agent using SSHADD at login.

SSHADD

Adds identities for the authentication agent.

```
SSHADD [OPTIONS] [FILE[,FILE,FILE]]
```

DESCRIPTION

SSHADD adds identities to SSHAGENT, the authentication agent. When run without arguments, SSHADD adds the file [.SSH]IDENTITY. Alternative file names can be given on the command line. If any file requires a passphrase, SSHADD asks for the passphrase from the user.

The authentication agent must be running and must have been executed by the user for SSHADD to work.

“File” is an identity or certificate file. If no file is specified, the files in the users[.SSH2] directory are used.

OPTIONS

/HELP	Display help text.
/LIST	List all identities currently represented by the agent.
/LOCK	Lock the agent with a password.
/NOSSH1	Agent cannot use SSH1 keys.
/PURGE	Remove all identities from the agent.
/REMOVE	Remove the identity from the agent. In order to remove identities, you must either issue the command from the subdirectory that the identities are located in, or issue the command using the full path name of the identity (as is seen in an SSHADD /LIST command).
/TIMEOUT=*n*	Agent should delete this key after the timeout value (in minutes) expires.
/UNLOCK	Unlock the locked agent.
/URL	Give key to the agent as a URL.

FILES

These files exist in SYS$LOGIN:

[.SSH]IDENTITY	Contains the RSA authentication identity of the user. This file should not be readable by anyone but the user. It is possible to specify a passphrase when generating the key. That passphrase is used to encrypt the private part of this file. This is the default file added by SSHADD when no other files have been specified. If SSHADD needs a passphrase, it reads the passphrase from the current terminal if it was run from a terminal. If SSHADD does not have a terminal associated with it but DECW$DISPLAY is set, it opens an X11 window to read the passphrase.
[.SSH]IDENTITY.PUB	Contains the public key for authentication. The contents of this file should be added to [.SSH]AUTHORIZED_KEYS on all systems where you want to log in using RSA authentication. There is no need to keep the contents of this file secret.
[.SSH]RANDOM_SEED	Seeds the random number generator. This file should not be readable by anyone but the user. This file is created the first time the program is run, and is updated every time SSHKEYGEN is run.

CERTTOOL

```
certtool [options] /pk10 /subject=<subject> /key_usage=<flags>
                   /extended_key_usage=<flags>
certview [options] /pk12 /input_files=<objects>
```

Description

The CERTTOOL utility is used for different needs concerning X.509 certificates.

Valid Options

/BITS=*n*	Key strength in bits (default 2048)
/DEBUG=*n*	Set debug level to *n*
/EXTENDED_KEY_USAGE =(flag1...flagn)	(PKCS#10 only) Extended key usage flags, as a comma-separated list. Valid values are: • anyExtendedKeyUsage • ServerAuth • clientAuth • codeSigning • emailprotection No extended flags are set by default.
/HELP[=(PK10,PK12)]	Display help. More detailed help on manipulating PKCS#10 and PKCS#12 certs is available by adding the PK10 and PK12 qualifier, respectively, to the HELP switch.
/INPUT_FILES=*(file1...filen)*	(PKCS#12 only) List of files to include in the PFX package.
/KEY_TYPE=*type*	Create a new key of type DSA or RSA.

/KEY_USAGE=*(flag1...flagn)*	(PKCS#10 only) Key usage flags, as a comma-separated list. Valid values are: • digitalSignature • nonRepudiation • keyEncipherment • dataEncipherment • keyAgreement • keyCertSign • CRLSign • encipherOnly • decipherOnly Default values are digitalSignature and keyEncipherment.
/OPTION=(*x*,*y*)	Set certificate option *x* to *y*. The options that can be set are dependent upon the type of certificate (PKCS#10 or PKCS#12) being affected. For PKCS#10: • DNS - set certificate DNS names • Email - set certificate email addresses For PKCS#12: • KeyPBE - set the PBE scheme for shrouding keys. “default” means pbeWithSHAAnd3-KeyTripleDES-CBC. • SafePBE - set the PBE scheme for protecting safes. “default” means pbeWithSHAAand40BitRC2-CBC.
/OUTPUT_FILE=*prefix*	Use *prefix* as the prefix for all output filenames. Private key filenames will be *prefix*.SSH2 and PKCS#10 files will be *prefix*.PKCS10.
/PRIVATE_KEY=*keyname*	Use *keyname* as the private key.
/SUBJECT=*”subject”*	(PKCS#10 only) Use *subject* as the certificate subject.
/VERSION	Display the version of CERTTOOL.

Example:

```
$ CERTTOOL /PK10 /SUBJECT=("cn=john doe,cn=lima,cn=beans”-
$_ /PRIVATE_KEY=DKA0:[JOHENDOE.SSH2]ID_DSA_1024_A
PKCS#10 creation successful.
Wrote certificate request to output.pkcs10.
```

CERTVIEW

```
certview [options] certificate [, certificate, ..., certificate]
```

Description

CERTVIEW can be used to view certificates and check their validity. This tool can also be used to output the data in format that is suitable for insertion in the SSH2_DIR:SSHD2_CONFIG configuration file.

Valid Options

/COMMENT	Prepend information lines with "#" (comment mark)
/DEBUG=*n*	Set debug level to *n*
/FORMAT_OUTPUT	Output data in a format suitable for insertion to *user-map*
/HELP	Display help
/QUIET	Don't display certificate information
/VALIDATE=*certificate*	Validate using the CA certificate *certificate*
/VERBOSE	Increase verbosity (display extensions).
/VERSION	Display version information

Example:

```
$ CERTVIEW MYCERT_PKCS7.P7B-1_SSH2_CRT
Certificate MYCERT_PKCS7.P7B-1_SSH2_CRT
Certificate issuer .......... : MAILTO=foo@bar.com, C=US, ST=CO, L=Colorado Springs, CN=FOOCA
Certificate serial number .... : 20668029027158235697617769792662904421
Certificate subject ......... : MAILTO=foo@bar,com, C=US, ST=CO, L=Colorado Springs, CN=FOOCA
```

CMPCLIENT

```
cmpclient [options]/ca_access_url=“<url>”/subject=“<subject>”cert-file [private-
key]
```

Description

Allows users to enroll certificates. It will connect to a CA (certification authority) and use 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.

Command Parameters

url	Specifies the URL for the Certification Authority
Subject	Specifies the subject name for the certificate. For example, “c-ca,o=acme,ou=development,cn=Bob Jones”
Cert file	Specifies the file the certification is written to.
Private key	Specifies the private key to be written to.

Valid Options

/BASE=<*name*>	Specify base prefix for the generated files.
/BITS=*n*	Specify the key length in bits.
/CA__URL=“<*url*>”	Specify the URL of the Certification Authority.
/DEBUG=*n*	Set debug to level *n* (0-60).
/ENROLLMENT_PROTOCOL =*prot*	Use specified enrollment protocol (SCEP or CMP).
/EXTENSIONS	Enable extensions in the subject name.
/GENERATE_KEY	Generate a new private key.
/HELP	Print this help text.

/PROXY_URL“=<*url*>”	Specify the URL of the HTTP proxy server URL to be used when connecting to the certification authority.
/REFNUM=*refnum:key*	Specify the CMP enrollment reference number and key.
/SOCKS_SERVER=“<*url*>”	Specify the URL of the SOCKS server URL to be used when connecting to the certification authority.
/SUBJECT=“<*subject*>”	Specifies the subject name for the certificate.
/TYPE=*rsa|dsa*	Specify the key type to generate (default: RSA)
/USAGE_BITS=*n*	Specify the key usage bits.
/VERSION	Print the version information for this program.

Examples:

1 Enroll a certificate and generate a DSA private key:

```
$ cmpclient/type=dsa/generate_key/base=mykey/refnum=1234:abc -
_$ /ca_access_url=“http://www.ca-auth.domain:8080/pkix/”-
_$ /subject=“c=us,o=foobar,cn=Dilbert Dogbert” ca-certification.crt
```

This will generate a private key called mykey.prv and a certificate called mykey-0.crt.

2 Enroll a certificate using a supplied private key and provide an e-mail extension:

```
$ cmpclient/base=mykey/refnum=12345:abcd -
_$ /ca_access_url=“http://www.ca-auth.domain:8080/pkix/”-
_$ /subject=“c=us,o=foobar,cn=Dilbert Dogbert:email=foo@bar.com” -
_$ ca-certification.crt my_private_key.prv
```

This will generate and enroll a certificate called mykey-0.crt.

Note! SSH stores and uses software certificates in DER encoded binary format. You can use *sshkeygen* to import and convert PKCS#12 packages (/pkcs_convert=file) into private key/certificate pair, X.509 format private key into SSH private key (/x509_convert=file) or PKC#7 into certificate (/extract_certs=file).

Public-key Subsystem

The public-key subsystem and assistant that can be used to add, remove and list public keys stored on a remote server. The public key assistant and server are based upon a recent IETF draft, so other implementations of SSH may not yet offer this functionality.

The Publickey assistant can be started with:

```
$ PUBLICKEY_ASSISTANT [qualifiers] [[user@]host[#port]]
```

Publickey Assistant Commands

ADD *key file name* - Transfers the key file name to the remote system. The file name specified is expected to be in the SSH2_CONFIG directory from the user's login directory. e.g., ADD ID_DSA_1024_A.PUB will transfer the public key in ID_DSA_1024_A.PUB to the remote system and updates the AUTHORIZATION. file on the remote system to include this key name.

CLOSE - Closes the connection to the remote system

DEBUG {*no* | *debug_leve*l} - Sets debug level (like in SFTP2)

DELETE *key finger-print*- Deletes the key that matches the fingerprint specified. It is necessary to do a LIST command before this to get a list of the finger prints (and for the program to build its internal database mapping fingerprints to keys).

EXIT- Exits the program.

HELP - Displays a summary of the commands available

LIST- Displays the fingerprint and attributes of keys stored on the remote system. The attributes that are listed will vary with key.

Example Output:

```
Fingerprint: xozil-bemup-favug-fimid-tohuk-kybic-huloz-fukuc-kuril-gezah-loxex

key type: ssh-dss

Comment: 1024-bit dsa, user@simple.example.com, Wed Jun 05 2012 21:05:40
```

OPEN [*user@]host[#port*] Opens a connection to a remote publickey subsystem.

QUIT- Quits the program.

UPLOAD *key file name* - Transfers the key file name to the remote system. The file name specified is expected to be in the SSH2_CONFIG directory from the user's login directory. e.g., ADD ID_DSA_1024_A.PUB will transfer the public key in ID_DSA_1024_A.PUB to the remote system and updates the AUTHORIZATION. file on the remote system to include this key name.

VERSION [*protocol version*] - Displays or sets the protocol version to use. The protocol version can only be set before the OPEN command is used. The default version is 1.

Publickey Assistant Qualifiers

/BATCHFILE - Provides file with publickey assistant commands to be executed. Starts SSH2 in batch mode. Authentication must not require user interaction.

/CIPHER - Selects encryption algorithm(s).

/COMPRESS - Enables SSH data compression.

/DEBUG - Sets debug level (0-99).

/HELP - Displays a summary of the qualifiers available.

/MAC - Selects MAC algorithm(s). /MAC=(*mac-1*,...,*mac-n*)

/PORT - Tells sftp2 which port sshd2 listens to on the remote machine.

/VERBOSE - Enables verbose mode debugging messages.
Equal to "/debug=2". You can disable verbose mode by using “debug disable.”

/VERSION - Displays version number only.

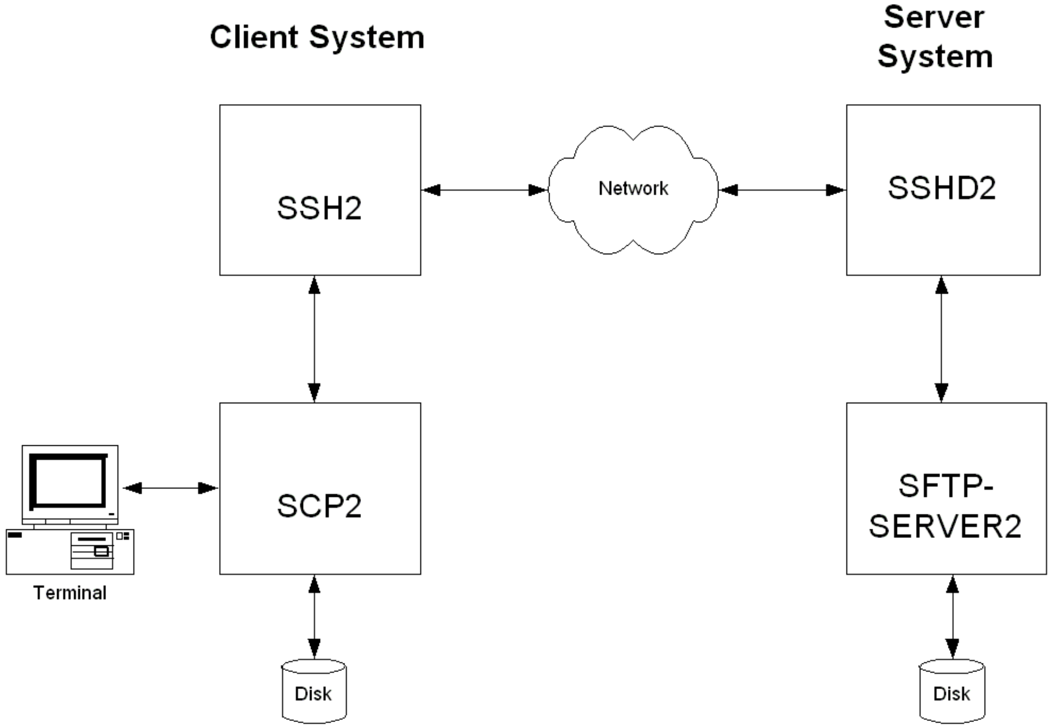
Other Implementations

VanDyke includes this in their SecureFX and VShell products. VanDyke also has a patch available for a server for OpenSSH.

Chapter 17 Secure File Transfer

There are three methods to do secure file transfer: SCP2, SFTP2, and FTP over SSH2. SCP2 and SFTP2 communicate with SSH2 for authentication and data transport (which includes encryption) to remote systems and to activate the SFTP-SERVER2 image. An SCP1 server is provided for compatibility with OpenSSH SCP.

The following diagram illustrates the relationship among the client and server portions of an SCP2 or SFTP2 file transfer:



SCP file transfers are different from FTP file transfers. With FTP a file can be transferred as ASCII, BINARY, RECORD, or in OpenVMS format (if MultiNet or TCPware is in use). In SCP the primary transfer format is BINARY. Also, the defined syntax for a file specification is UNIX syntax. Due to these restrictions, files that are transferred from dissimilar systems may or may not be useful. ASCII transfers are done by searching the transferred data for the specified newline sequence and making the specified substitution. Process Software has used methods available in the protocol to attempt to improve the chances that files will be useful upon transfer. The SSH File Transfer Protocol is an evolving specification, and some implementations may not support all options available in the protocol, or worse, not tolerate some optional parts of later versions of the protocol.

Process Software has used the defined extensions in the protocol to transfer information about the VMS file header characteristics such that when a file is transferred between two VMS systems running MultiNet v4.4 or higher, TCPware v5.6, and/or SSH for OpenVMS, 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 text file is transferred to a non-VMS system, a method has been provided to convert those files that can be translated into a format that will be usable on the remote system. Files that are converted from non-VMS systems are stored as stream files on the VMS system, which provides compatibility for text files from those systems. Filenames are SRI-encoded when files are stored on ODS-2 disks.

SCP-SERVER1

The SCP-SERVER1 program is used when a system with OpenSSH initiates an SCP command. OpenSSH uses RCP over SSH2 instead of the SFTP protocol. SCP-SERVER1 will always convert VMS text files (if possible) when copying a file from VMS. Converted VMS text files may have some trailing nulls at the end of them, due to the RCP protocol not being able to tolerate a file that comes up short of the reported size. SCP-SERVER1 (and SFTP-SERVER2) use sophisticated methods to estimate the amount of user data in the file to minimize this. On ODS-5 disks the estimation routine uses the file size hint if it is valid. On ODS-2 disks (and ODS-5 without a valid size hint), the size of the file and file characteristics are used to estimate the amount of user data. The method provides as accurate an estimate as possible without actually reading the file and never underestimates the amount of data in the file. Underestimating would cause significant problems as the programs use the size of the file to determine how much data to expect.

SCP2

Usage

```
SCP2 [qualifiers] [[user@]host[#port]::]file [[user@]host[#port]::]file
```

Note! The source and destination file specification must be quoted if they contain a user specification or a non-VMS file specification.

Qualifiers

Table 17-1 SCP Qualifiers

Qualifier	**Description**
/ASCII[=*newline convention*]	Newline convention is one of dos, mac, unix, vms, or sftp. The newline convention specified is the newline convention to use if a newline convention is not specified by the server. Allowed values: dos (\r\n), mac (\r) , unix (\n), vms (\n) , sftp (\r\n). Default = unix.
/BATCH	Starts SSH2 in batch mode. Authentication must be possible without user interaction.
/BUFFER_SIZE=*integer*	Number of bytes of data to transfer in a buffer. Default is 7500. Minimum value is 512.
/CIPHER=*(cipher-1,...,cipher-n)*	Selects an encryption algorithm(s).
/COMPRESS	Enables SSH data compression.
/CONCURRENT_REQUEST=*integer*	Number of concurrent read requests to post to the source file. Default is 4.
/DEBUG=*level*	Sets a debug level. (0-99)
/DIRECTORY	Forces the target to be a directory.
/HELP	Displays the help text.
/IDENTITY_FILE=*file*	Identifies the file for public key authentication.
/OVERWRITE	Overwrite existing file instead of deleting first.
/PORT=*number*	Tells SCP2 which port SSHD2 listens to on the remote machine.
/PRESERVE	Preserves file attributes and timestamps.
/NOPROGRESS	Does not show progress indicator.
/QUIET	Does not display any warning messages.
/RECORD	Open the source file in VMS Record mode if possible. This is equivalent to record mode transfer in SFTP2. The file is transferred as a stream of records with no carriage control added between them.
/RECURSIVE	Processes the entire directory tree.
/REMOVE	Removes the source files after copying.
/TRANSLATE_VMS= *(ALL, NONE, VARIABLE, FIXED, VFC)*	Selects the VMS text files to be translated (default=ALL). Note that /ASCII performs a similar function and may be supported in other SCP products.

/VERBOSE	Displays verbose debugging messages. Equal to "/debug=2".
/VERSION	Displays the version number only.
/VMS	Negotiates the ability to transfer VMS file information.

Note! /ASCII, /VMS and /TRANSLATE_VMS are mutually exclusive

File Specifications

The source and destination strings are changed to lowercase unless they are enclosed in quotes, in which case they are left the same. File specification must be in UNIX format for remote systems, unless the remote system is running TCPware 5.6, MultiNet v4.4 or higher, or SSH for OpenVMS, and /VMS or /TRANSLATE_VMS (source files only) are used. UNIX format file specifications need to be enclosed in quotes (") if they contain the / character to prevent the DCL parsing routines from interpreting the string as a qualifier.

Qualifiers

/ASCII[=newline convention]

Uses the newline convention specified if the server does not specify a newline convention. Available conventions are: dos (\r\n), mac (\r) , unix (\n), vms (\n) , sftp (\r\n). Default = unix.

/BATCH

Starts SSH2 in BATCH mode. When SSH2 is running in BATCH mode it does not prompt for a password, so user authentication must be performed without user interaction.

/BUFFER_SIZE=integer

Number of bytes of data to transfer in a buffer. Default is 7500.

/CIPHER=(cipher,...,cipher-n)

Lets you select which SSH2 cipher to use.

/COMPRESS

Enables SSH2 data compression. This can be beneficial for large file transfers over slow links. The compression level is set by the client configuration file for SSH2.

/CONCURRENT_REQUEST=integer

Number of concurrent read requests to post to the source file. Default is 4.

/DEBUG

Enables debugging messages for SCP2 and SSH2. Higher numbers get more messages. The legal values are between 0 (none) and 99. Debugging for SFTP-SERVER2 is enabled via the TCPWARE_SSH_SFTP_SERVER_DEBUG logical.

/DIRECTORY

Informs SCP2 that the target specification should be a directory that the source file(s) will be put in. This qualifier is necessary when using wildcards in the source file specification, or /RECURSIVE.

/HELP

Displays command qualifier list and parameter format.

/IDENTITY_FILE=file

Specifies the identity file that SSH2 should use for Public-Key authentication.

/PORT=number

Specifies the port that SSH2 uses on the remote system. Note that if both the source and destination files are remote, this value is applied to both. If SSH2 is available on different ports on the two systems, then the #port method must be used.

/PRESERVE

Sets the Protection, Owner (UIC), and Modification dates on the target file to match that of the source file. The adjustment of timestamps for timezones is dependent upon the logical SYS$LOCALTIME being set correctly. This is defined automatically on VMS V7 and can be defined similarly on earlier versions of VMS. /PRESERVE is not very useful when the target machine is a VMS system as VMS does not provide runtime library calls for setting the file attributes (owner, protection) and timestamps. Note that the VMS modification date (not the creation date) is propagated to the remote system. When files are copied between two VMS systems and /VMS is used /PRESERVE is implied and the process of transferring VMS attributes preserves the information about the protection, dates, and file characteristics.

/NOPROGRESS

SCP2, by default, updates a progress line at regular intervals when it is run interactively to show how much of the file has been transferred. This qualifier disables the progress line.

/QUIET

Disables warning messages. Note that it does not disable warning messages from SFTP-SERVER2, which return on the error channel.

/RECORD

Open the source file in VMS record mode. This copies the source file to the destination as records converted to a stream of bytes without any carriage control between records. This is equivalent to RECORD mode transfer in SFTP.

/RECURSIVE

Copies all of the files in the specified directory tree. Note that the top level directory on the local system is not created on the remote system. Only the most recent version is copied unless in VMS mode and the TCPWARE_SFTP_VMS_ALL_VERSIONS logical is defined to be TRUE.

/REMOVE

Deletes the source files after they have been copied to the remote system.

/TRANSLATE_VMS

Translates VMS text files in the copying process to byte streams separated by linefeeds because the defined data transfer format for SCP2 is a binary stream of bytes.

/TRANSLATE_VMS is only applicable to the source specification. If a remote source file is specified, then that system must be running MultiNet v4.4 or higher, TCPware 5.6, or SSH for OpenVMS. If /TRANSLATE_VMS is specified with no value, then VARIABLE, FIXED, and VFC (Variable, Fixed Control) files are translated to stream linefeed files. If the value is NONE, no files are translated. VARIABLE, FIXED, and VFC can be combined in any manner. The SFTP-SERVER2 process uses the value of the logical TCPWARE_SFTP_TRANSLATE_VMS_FILE_TYPES to determine which files should be translated automatically. This is a bit mask with bit 0 (1) = FIXED, bit 1 (2) = VARIABLE, and bit 2 (4) = VFC. These values can be combined into a number between 0 and 7 to control which files are translated.

Note! Due to the structure of the programs, the SCP2 program uses this logical if the /TRANSLATE_VMS qualifier has not been specified.

/VERBOSE

Displays debugging messages that allow the user to see what command was used to start up SSH and other basic debugging information. Note that debugging information can interfere with the normal display of the progress line. Equivalent to /DEBUG=2.

/VERSION

Displays the version of the base SCP2 code.

/VMS

Transfers VMS file information similar to that transferred in OVMS mode in FTP such that VMS file structure can be preserved. All of the information transferred in FTP OVMS mode is transferred along with the file creation date and protection. Timestamps are not adjusted for timezone differences in VMS transfers. If the file is a contiguous file, and it is not possible to create the file contiguously, and the logical TCPWARE_SFTP_FALLBACK_TO_CBT has the value of TRUE, YES, or 1, SFTP-SERVER2 attempts to create the file Contiguous, Best Try. VMS mode is only available with SCP2 provided in MultiNet v4.4 or higher, TCPware 5.6, and SSH for OpenVMS.

The logical name TCPWARE_SCP2_VMS_MODE_BY_DEFAULT can be defined to TRUE, YES, or 1 to specify that /VMS should be the default unless /NOVMS or /TRANSLATE_VMS are specified. /VMS and /TRANSLATE_VMS cannot be used on the same command line. If /VMS is not specified, but the logical is set to enable it by default, a /TRANSLATE_VMS on the command line will take precedence.

Note that even though SCP2 & SFTP-SERVER2 pass the request for VMS file transfers or to translate a VMS file in a manner that is consistent with the protocol specification, other implementations may not handle this information well. Since there is no error response present at that point in the protocol, the program hangs. To prevent it from hanging forever, the logical TCPWARE_SCP2_CONNECT_TIMEOUT is checked to see how long SCP2 should wait for a response when establishing the connection. The format for this logical is a VMS delta time. The default value is 2 minutes. If SCP2 times out before a connection is established with SFTP-SERVER2 and /VMS or /TRANSLATE_VMS were specified, a warning message is displayed and the initialization is tried again without the request for VMS information (or /TRANSLATE_VMS). This retry is also subject to the timeout, and if the timeout happens again, then SCP2 exits. This helps for implementations that ignore the initialization message when information they do not recognize is present; implementations that abort will cause SCP2 to exit immediately.

Logicals

For the following logicals, all that start TCPWARE_SFTP apply to the SCP2 client, SFTP2 client and SFTP2 server.

TCPWARE_SFTP_FALLBACK_TO_CBT

When defined to TRUE, YES, or 1 (the number 1) and a VMS file transfer is being performed, this logical creates a Contiguous file if that file has Contiguous characteristics. The file will be created as Contiguous Best Try if there is insufficient space to create it as Contiguous.

TCPWARE_SFTP_TRANSLATE_VMS_FILE_TYPES

This is a bit mask that determines which VMS file types should be translated when not operating in VMS mode.

- Bit 0 (1) = FIXED
- Bit 1 (2) = VARIABLE
- Bit 2 (4) = VFC

The values are:

- 0 (zero) = NONE
- 7 = ALL

Note that this logical affects SCP2 as well as the server, as SCP2 has the server built into it for handling local file access. If this logical is not defined, the value 7 will be used.

TCPWARE_SCP2_CONNECT_TIMEOUT

This logical defines a number specifying how long SCP2 should wait for a response to the INITIALIZE command from the server program. This is a VMS delta time number. The default is 2 minutes.

TCPWARE_SCP2_VMS_MODE_BY_DEFAULT

When defined to TRUE, YES, or 1, this logical chooses the /VMS qualifier if /TRANSLATE_VMS or /NOVMS has not been specified.

TCPWARE_SFTP_RETURN_ALQ

When defined to TRUE, YES, or 1 (the number one) and files are being transferred in VMS mode, this logical includes the Allocation Quantity for the file in the file header information. This is disabled by default because copying a small file from a disk with a large cluster size to a disk with a small cluster size causes the file to be allocated with more space than necessary. You have the option of retaining the allocated size of a file if it was allocated the space for a reason. Some combinations of file characteristics require that the Allocation Quantity be included in the file attributes; this is handled by SCP2/SFTP-SERVER2.

TCPWARE_SSH_SCP_SERVER_DEBUG

Enables debugging messages for the SCP-SERVER1 image that provides service to SCP commands that use the RCP over SSH2 protocol (OpenSSH). When this is defined, the file
SCP-SERVER.LOG is created in the user's login directory. These files are not purged. Larger values yield more debugging information.

TCPWARE_SSH_SFTP_SERVER_DEBUG

Enables debugging messages for the SFTP-SERVER2 image that provides service to SCP2 commands that use the SFTP protocol. When this is defined, the file SFTP-SERVER.LOG is created in the user's login directory. These files are not purged. Larger values yield more debugging information

TCPWARE_SFTP_MAXIMUM_PROTOCOL_VERSION

This logical can be used to limit the version of the SSH File Transfer Protocol that the SFTP client and Server use. This can sometimes provide a work-around for problems encountered with different implementations of the protocol. The default value is 4. Protocol versions 2 and 3 are also used by popular implementations.

TCPWARE_SFTP_VMS_ALL_VERSIONS

This logical controls whether or not all versions of a file are returned. The values TRUE, YES or 1 (the number one) will cause all versions to be returned, any other value is to only return the name of the file without a version. The default is to return only one filename without the version number.

TCPWARE_SFTP_NEWLINE_STYLE

This logical controls the newline style that SFTP uses. Which can be helpful in transferring text files. The values are: UNIX <lf>, VMS <lf>, MAC <cr>. If the logical is not defined, or defined to any other value, then <cr><lf> will be used for the text line separator as documented in the SSH File Transfer specification.

TCPWARE_SFTP_CASE_INSENSITIVE

This logical causes SFTP to treat filenames in a case insensitive manner when it is defined to TRUE, YES, or 1 (the number one).

TCPWARE_SFTP_ODS2_SRI_ENCODING

This logical controls whether or not SRI encoding is used for filenames on VMS ODS-2 disks. If the logical is not defined, or is defined to TRUE, YES, or 1 (the number one) then SRI encoding is used on ODS-2 disks for filenames that contain uppercase letters and special characters.

TCPWARE_SFTP_FILE_ESTIMATE_THRESHOLD

This logical controls the minimum number of blocks that a text file must be for an estimated transfer size to be returned instead of an exact size. The default is to estimate the transfer size for all text files.

TCPWARE_SFTP_DEFAULT_FILE_TYPE_REGULAR

If this logical is defined to TRUE, YES or 1 (the number one), then the SFTP server will use a default file type of REGULAR instead of UNKNOWN for OPEN operations. This can correct problems with filenames without a . (dot) in them getting .dir added to them. The filename will appear with a . at the end of the name in directory listings.

TCPWARE_SFTP_<username>_CONTROL

The logical TCPWARE_SFTP_<username>_CONTROL can be defined /SYSTEM to any combination of NOLIST, NOREAD, NOWRITE, NODELETE, NORENAME, NOMKDIR, NORMDIR, to restrict operations for the username in the logical. NOWRITE will disable PUT, DELETE, RENAME, MKDIR, RMDIR; NOREAD will disable GET and LIST.

TCPWARE_SFTP_<username>_ROOT

The logical TCPWARE_SFTP_<username>_ROOT can be defined /SYSTEM to restrict the user to the directory path specified. Subdirectories below the specified directory are allowed.

SSH_SFTP_LOG_SEVERITY

The logical SSH_SFTP_LOG_SEVERITY can be defined /SYSTEM to 20000 to log file transfers or 30000 to log all SFTP operations.

SSH2_SFTP_LOG_FACILITY

The logical SSH2_SFTP_LOG_FACILITY must also be defined /SYSTEM to specify the logging class that is used with OPCOM.

Values below 5 will use the network class; 5 will use OPER1, 6 will user OPER2, etc. The maximum value that can be specified is 12, which will use OPER8.

TCPWARE_SFTP_SEND_VENDOR_ID

If this logical is defined to “No”, “False” or “0” (zero), then the SFTP2 client will not send the extended command containing the vendor-id upon completion of version negotiation with the server.

SFTP2

File Specifications

File specification must be in UNIX format for remote systems, unless /VMS transfers are being used.

SFTP2 Command Syntax and Qualifiers

Usage

```
SFTP2 [qualifiers] [[user@]host[#port]]
```

If the username@ is included in the remote system specification, the specification must be enclosed in quotes.

Qualifiers

Table 17-2 SFTP2 Qualifiers

Qualifier	**Description**
/BATCHFILE=*<file specification>*	Provides file with SFTP commands to be executed. Starts SSH2 in batch mode. Authentication must not require user interaction.
/BUFFER_SIZE=*integer*	Number of bytes of data to transfer in a buffer. Default is 7500.
/CIPHER=*(cipher-1,...,cipher-n)*	Selects encryption algorithm(s).
/COMPRESS	Enables SSH data compression.
/CONCURRENT_REQUEST=*integer*	Number of concurrent read requests to post to the source file. Default is 4.
/DEBUG=*level*	Sets debug level (0-99).
/HELP	Displays help.
/MAC=*(mac-1,...,mac-n)*	Select MAC algorithm(s).
/NOPROGRESS	Do not show progress indicator.
/PORT	Tells SFTP2 which port the SSHD2 server is listening on.
/VERBOSE	Enables verbose mode debugging messages. Equal to "/debug=2".

	You can disable verbose mode by using “debug disable.”
/VERSION	Displays version number only.
/[NO]VMS	Negotiates ability to transfer VMS file information. VMS transfer mode will be automatically negotiated if SFTP2 detects that the server is capable of doing VMS transfers unless /NOVMS is specified.

SFTP2 Commands

Table 17-3 SFTP2 Commands

SFTP2 Command	**Description**
ASCII[{-*s* |<*remote*> [<*local*>]}]	With “-*s*” option, shows current newline convention. <*remote nl conv*> sets remote newline convention. <*local nl conv*> operates on local side, but is not as useful (the correct local newline convention is usually compiled in, so this is mainly for testing). You can set either of these to “ask”, which will cause sftp to prompt you for the newline convention when needed. With the exception of “-*s*” option, this command sets transfer mode to ascii. Available conventions are “dos”, “unix”, “sftp”, “vms”, or “mac”, using “\r\n”, “\n”, “\r\n”, “\n” and “\r” as newlines, respectively. Note that some implementations of SFTP may check to see if a file can be transferred in ASCII mode before doing so, and return errors for files that cannot be transferred. SSH for OpenVMS, MultiNet, and TCPware make this check.
AUTO	Sets the transfer mode (ASCII or BINARY) to depend upon the extension of the file specification.
BINARY	Sets the transfer mode to be binary. (This is the default.)
BUFFERSIZE *[number]*	Sets the size of the buffer used for file transfer. A larger buffer size helps speed large transfers. Displays the current buffer size when no parameter is specified.
CD <*directory specification*>	Changes current directory on remote system. VMS file specifications may be used when operating in VMS mode. A logical name must include the trailing colon so that it can be recognized as such. SFTP from other vendors cannot use VMS specifications due to the way that SFTP works.

CHMOD [-R] <mode> file [file...]	Change the protection on a file or directory to the specified octal mode. (Unix values). -R recourses over directories.
CLOSE	Closes connection to the remote server.
DEBUG {disable | no |<*debug level>}*	Sets the debug level for SFTP2. It does not change the current debug level for SSH2 for an existing connection, but will be used with SSH2 for a new connection. With “disable” or “no”, this disables all debugging current sessions for SFTP2.
DELETE <*file specification*>	Removes the specified file from the remote system.
DIRECTORY [<*file |directory specification>]*	Displays the contents of the current directory or specified directory in VMS format when the transfer mode is VMS. File names are displayed as they would be with a DIRECTORY command from DCL.
EXIT	Exits SFTP client.
GET [--preserve-attributes | -p] <*file*1> [<*file2*>...]	Retrieves the specified file(s) from the remote system and stores it in the current working directory on the local system. File names are case sensitive and in UNIX format. When operating in VMS mode, either UNIX or VMS-style file specifications can be used. Directories are recursively copied with their contents. Multiple files may be specified by separating the names with spaces. If --preserve-attributes or -p is specified, then SFTP attempts to preserve timestamps and access permissions. Note that a target filename cannot be provided.
GETEXT	Displays the list of file extensions to use ASCII transfers when in AUTO mode. The initial value is txt,htm*,pl,php*
HELP	Displays help on commands.
LCD <*directory specification*>	Changes the current directory on the local system. VMS file specifications may be used when in VMS mode.
LCHMOD [-R] <mode> file [file...]	Change the protection on a file or directory on the local connection to the specified octal mode. (Unix values). -R recourses over directories.
LCLOSE	Close the local connection.

LDELETE<*file*>	Removes the specified file from the local system. VMS file specifications may be used when in VMS mode.
LDIRECTORY [<*file* |*directory specification*>]	Displays the contents of the current directory for the local system in VMS format when the transfer mode is VMS. File names are displayed as they would be with a DIRECTORY command from DCL.
LLS [<*file* |*directory specification*>]	Displays the contents of the current directory or specified directory in UNIX format. Lists the names of files on the local server. For directories, contents are listed. See LS for options and more details.
LLSROOTS	Like LSROOTS, but for the “local” side.
LMKDIR <*directory specification*>	Creates the specified directory on the local system.
LOCALOPEN {[*user@host*[*#port*] | -1}	Tries to connect the local side to the host <*hostname*>. If successful, “lls” and friends will show the contents of the filesystem on that host. With the -l option, connects to the local filesystem (which doesn’t require a server). There is an implied LOCALOPEN *-l* when SFTP2 starts up. Note that an implicit LOCALOPEN is done when SFTP2 starts, so the only time that a user needs to do a LOCALOPEN is when neither directory tree is immediately accessible. OPEN is the command that is generally used to establish the connection with the remote system. LOPEN is a synonym for LOCALOPEN.
LPWD	Displays the current working directory on the local system.
LREADLINK <*path*>	Provided that <*path*> is a symbolic link, shows where the link is pointing to. This command is not supported for VMS.
LRENAME <*oldfile*><*newfile*>	Renames a file on the local system.
LRM <*file specification*>	Removes the specified file from the local system. VMS file specifications may be used when in VMS mode.
LRMDIR <*directory specification*>	Deletes a directory on the local system.

LS *[-R] [-l] [-S] [-r] [<file> ...]*	Displays the contents of the current directory or specified directory in UNIX format. Lists the names of files on the remote server. For directories, contents are listed. When the *-R* is given, directory trees are listed recursively. (By default, subdirectories of the arguments are not visited.) When the *-l* option is given, permissions, owners, sizes, and modification times are also shown. When the -S options is specified sorting is based upon file size instead of alphabetically. The -r option reverses the sort order. When no arguments are given, it assumes that the contents of "." (current working directory) are being listed. Currently, the options *-R* and *-l* are mutually incompatible. Ls will fill a screen with output, then wait for the user to decide if they want more or have seen enough.
LSROOTS	Displays the virtual roots of the server. (This VanDyke Software's V Shell extension. Without this you can't know the filesystem structure of a V Shell server). This is also a VMS extension to display the roots (devices) on the VMS system. Though the commands are the same, the information provided is not compatible with what is displayed by VanDyke Software's Secure FX.
LSYMLINK *<targetpath><linkpath>*	Like SYMLINK, but for the "local" side.
MGET [--preserve-attributes | -p] <*file1*> [<*file2*>...]	Retrieves multiple files from the remote system and stores them in the current working directory on the local system. If --preserve-attributes or -p is specified, then SFTP attempts to preserve timestamps and access permissions.
MKDIR <*directory specification*>	Creates the specified directory on the remote system.
MPUT [--preserve-attributes | -p] <*file1*> [<*file2*>...]	Stores multiple files in the current working directory on the remote system. File names are case-sensitive and in UNIX format. When operating in VMS mode, either UNIX or VMS-style file specifications can be used. Directories are recursively copied with their contents. Multiple files may be specified by separating the names with spaces. If --preserve-attributes or -p is specified, then SFTP attempts to preserve timestamps and access permissions.
OPEN *{-1| [user@]host[#port]}*	Tries to connect to the host<*hostname*>. Or with the -l option, connects the remote side to the local filesystem (which doesn't require a server).

PUT [--preserve-attributes | -p] <file1> [<file2>...]	Stores the specified file in the current working directory on the remote system. File names are case-sensitive and in UNIX format. When operating in VMS mode, either UNIX or VMS-style file specifications can be used. Directories are recursively copied with their contents. Multiple files may be specified by separating the names with spaces. If --preserve-attributes or -p is specified, then SFTP attempts to preserve timestamps and access permissions. Note that a target filename cannot be provided.
PWD	Displays the current working directory on the remote system. Displayed in VMS format when in VMS mode; otherwise displayed in UNIX format.
QUIT	Exits SFTP client.
READLINK *<targetpath><linkpath>*	Provided that <*path*> is a symbolic link, shows where the link is pointing to. Not valid for VMS systems as VMS does not have symbolic links.
RECORD	Enters record transfer mode if the server supports Process Software's record open. The direction in which record transfer mode is possible will be displayed in response to this command. In record transfer mode the source file is opened as binary records and the destination file is opened as binary. This produces the same effect as TCPware's FTP server BINARY transfer when a BLOCK_SIZE has not been specified, and can be used to transfer a file that contains VMS records to a system that can only handle "flat" files.
RENAME *<oldfile><newfile>*	Renames file on the remote system.
RM *<file specification>*	Removes the specified file from the remote system.
RMDIR *<directory specification>*	Deletes a directory on the remote system.
SETEXT <*ext1*>[<*ext2*>...]	Sets the list of file extensions to use ASCII transfers when in AUTO mode. Individual file extensions must be separated by spaces.
STATUS	Shows the transfer mode, remote server name, and remote server version. The current newline sequence is displayed if operating in ASCII or AUTO mode.

SYMLINK <*targetpath*><*linkpath*>	Creates symbolic link <*linkpath*>, which will point to <*targetpath*>. Not valid for VMS systems as VMS does not have symbolic links.
VERBOSE	Enables verbose mode (identical to "/DEBUG=2" command-line option). You may later disable verbose mode by "debug disable".
VMS	Sets the transfer mode to include VMS file information.

Logicals

The following logicals are specific to SFTP2.

TCPWARE_SFTP_VMS_MODE_BY_DEFAULT

When defined to TRUE, YES, or 1, this logical chooses the /VMS qualifier if /NOVMS has not been specified.

Configuration File Parameters

The system wide configuration file (SSH2_DIR:SSH2_CONFIG.) or the user's configuration file (SYS$LOGIN:[.SSH2]SSH2_CONFIG.) can be used to specify the following parameters. The user's configuration file takes precedence over the system configuration file.

Table 17-4 SFTP/SCP2 user configuration parameters

FilecopyMaxBuffers	This is equivalent to the /CONCURRENT_REQUEST qualifier on the SFTP2 or SCP2 command line. The command line qualifier will supersede any value in the configuration file.
FilecopyMaxBuffersize	This is equivalent to the SFTP2 BUFFERSIZE command or the SCP2 /BUFFER_SIZE qualifier. The command or qualifier takes precedence.

The system server configuration file (SSH2_DIR:SSHD2_CONFIG.) can include parameters to control which users can perform remove SSH commands (including SSH terminal sessions) as well as SFTP2 access.

Table 17-5 SSH2 terminal restriction parameters

Terminal.AllowUsers	Allow users in the specified list to create SSH2 terminals and do interactive commands
Terminal.DenyUsers	Prevent users in the specified list from creating SSH2 terminals and performing interactive commands. The users can still use the SFTP2, SCP1 and Public Key servers.

Terminal.AllowGroups	Allow groups in the specified list to create SSH2 terminals and do interactive commands
Terminal.DenyGroups	Prevent groups in the specified list from creating SSH2 terminals and performing interactive commands. The groups can still use the SFTP2, SCP1 and Public Key servers.

FTP over SSH

SSH2 can be used to set up port forwarding that can be used for FTP. This allows users to use the richness of the FTP command set to access files on a remote system and have their control and data information encrypted. The command format to set up the SSH port forwarding is:

```
$ ssh <remote_host_name>/local_forward=
(“ “ “ftp/<forwarded_port_number>:localhost:21” ” ”)
```

The usual SSH authentication mechanisms come into play, so there may be a request for a password and a terminal session is established to the remote host. As long as this terminal session is alive, other users on the local system can use FTP to access the remote system over an encrypted channel. The location of the quotes is important, as it is necessary to prevent DCL from interpreting the / in the local forwarding information as the start of a new qualifier, and SSH2 does not know or expect to find the ( ) around the forwarding information. Note that the “localhost” inside of the forwarding string is important, as it will make the connection to FTP on the remote system come from localhost, which will then allow FTP to open the data port.

When a user desires to use an encrypted FTP connection, the following sequence of commands would be issued:

```
PORT <forward_port_number>
OPEN LOCALHOST
```

Normal FTP authentication takes place and multiple FTP sessions may use a single forwarded port. The FTP protocol filter in SSH2 scans the FTP command stream for the FTP PORT and PASV commands and their replies, and makes substitutions in these commands and replies to use a secure data stream through the SSH2 session that has been set up. This command will establish an encrypted FTP session with the remote host that the SSH connection is sent to.

To allow a single system to act as a gateway between two networks, add /ALLOW_REMOTE_CONNECT to the SSH command that initiates the connection.

Appendix A References

Introduction

This appendix lists documentation to which you can refer for additional details about TCPware for OpenVMS, TCP/IP protocol suite, networking concepts, and related subjects.

TCPware for OpenVMS Documentation

Be sure you have the following additional TCPware for OpenVMS documents available for reference:

Installation & Configuration Guide

Management Guide

Network Control Utility (NETCU) Command Reference

Programmer's Guide

Requests for Comments (RFCs)

Requests for Comments (RFCs) documents contain the specifications for all internet protocols. Unless specifically noted otherwise on the RFC itself, all RFCs are for unlimited distribution.

You can obtain RFCs by going to the http://www.rfcs.org web site.

Table A-1 lists the RFCs containing the protocol specifications implemented by TCPware for OpenVMS.

Table A-1 A Subset of RFCs Implemented by TCPware for OpenVMS

Title	**RFC #**
User Datagram Protocol (STD 6)	768
Internet Protocol: DARPA Internet Program Protocol Specification	791
Internet Control Message Protocol (see also RFC 950)	792
Transmission Control Protocol	793

Simple Mail Transfer Protocol (STD 10)	821
Standard for the Format of 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
A Standard for the Transmission of IP Datagrams over Ethernet Networks	894
Reverse Address Resolution Protocol	903
Broadcasting Internet Datagrams (STD 5)	919
Broadcasting Internet Datagrams in the Presence of Subnets (STD 5)	922
Internet Standard Subnetting Procedures (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
A Standard for the Transmission of IP Datagrams over IEEE 802 Networks	1042
Internet Protocol on Network Systems HYPERchannel Protocol Specification	1044
A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP	1055
RPC: Remote Procedure Call Protocol Specification, 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 the Internet Protocol	1108
Host Extensions for IP Multicasting (STD 5)	1112
Compressing TCP/IP Headers for Low-Speed Serial Links	1144
Structure and Identification of Management Information...(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
Management Information Base for Network Management...	1213
Tunneling IPX Traffic through IP Networks	1234
BSD Rlogin	1282
The Finger User Information Protocol	1288
Network Time Protocol (Version 3) Specification, Implementation & Analysis	1305
TCP Extension for High Performance	1323
DNS NSAP RRs	1348
Type of Service in the Internet Protocol Suite	1349
The TFTP Protocol (Revision 2) (STD 33)	1350
Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode	1356
TELNET Remote Flow Control Option	1372
Transmission of IP and ARP over FDDI Network (STD 36)	1390
IP Multicast over Token-Ring Local Area Networks	1469
Encoding Header Field for Internet Messages	1505
Applicability Statement for the Implementation of CIDR	1517
An Architecture for IP Address Allocation with DICR	1518
Classless Inter-Domain Routing (CIDR):...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
Post Office Protocol - Version 3 (STD 53)	1939
Internet Message Access Protocol - Version 4rev1	2060
Dynamic Host Configuration Protocol	2131
DHCP Options and BOOTPD Vendor Extensions	2132
Dynamic Updates in the Domain Name System (DNS Update)	2136
Secure Domain Name System Dynamic Update	2137
Agent Extensibility (AgentX) Protocol Version 1	2741
Definitions of Managed Objects for Extensible SNMP Agents	2742

Internet, TCP/IP Protocol Suite, and Related Subjects

The following RFCs are also available on more general Internet, TCP/IP, and related subjects:

RFC 1118, *The Hitchhikers Guide to the Internet*

RFC 1359, *Connecting to the Internet: What Connecting Institutions Should Anticipate*

RFC 1392, *Internet Users' Glossary*

RFC 1432, *Recent Internet Books*

RFC 1462, *FYI on "What is the Internet?"*

RFC 1463, *FYI on Introducing the Internet—A Short Bibliography of Introductory Internetworking Readings for the Network Novice*

RFC 2151, *A Primer on Internet and TCP/IP Tools and Utilities*

The following books are particularly useful references:

Comer, Douglas E. [1995], *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture,* Third edition, Prentice-Hall.

Comer, Douglas E. & David L. Stevens [1994], *Internetworking with TCP/IP, Volume II: Design, Implementation, and Internals,* Second edition, Prentice-Hall.

Comer, Douglas E. & David L. Stevens [1996], *Internetworking with TCP/IP, Volume III: Client-Server Programming and Applications for the BSD Socket Version,* Second edition, Prentice-Hall.

Frey, Donnalyn, Rick Adams [1989], *A Directory of Electronic Mail, Addressing and Networks,* O'Reilly & Associates, Inc.

LaQuey, Tracy L. (editor) [1990], *The User's Directory of Computer Networks*, HP Press.

Perlman, Radia [1992], *Interconnections: Bridges and Routers,* Addison-Wesley.

Quarterman, John S. [1990], *The Matrix: Computer Networks and Conferencing Systems Worldwide,* HP Press.

Santifaller, Michael [1991], *TCP/IP and NFS: Internetworking in a UNIX Environment,* translated by Stephen S. Wilson, Addison-Wesley.

Stallings, William [1991], *Data and Computer Communications,* Third edition, MacMillan.

Stevens, W. Richard [1990], *UNIX Network Programming,* Prentice-Hall.

Tanenbaum, Andrew S. [1996], *Computer Networks,* Third edition, Prentice-Hall.

Tolhurst, William A. et al. [1994], *Using the Internet, Special Edition,* Que Corp.

Table A-2 lists documentation to which you can refer for details on specific topics.

Table A-2 Additional Documentation

For Details on...	**See...**
Process Software’s home page on the World Wide Web	For information about Process Software, its products, and its services, enter the following Universal Resource Locator (URL) from your World Wide Web browser: http://www.process.com/
VAX WAN	*VAX WAN Device Drivers Specifications* available from HP for details on the device drivers TCPware for OpenVMS supports (DSV11, DSB32, and DST32).
DECwindows	*VMS DECwindows User’s Guide* and the *VMS DECwindowsMotif User’s Guide* available from HP.
Domain Name Services (DNS)	Albitz, Paul & Cricket Liu, *DNS and Bind*, O’Reilly Associates.
Dynamic Host Configuration Protocol (DHCP)	Droms, Ralph and Ted Lemon, *TheDHCP Handbook: Understanding, Deploying, and Managing Automated Configuration Services,* 1999 Macmillan Technical Publishing, 201 West 103rd Street, Indianapolis, IN 46290 ISBN 1-57870-137-6

Ethernet	*Ethernet Data Link Layer and Physical Layer Specifications* available from HP or from your Ethernet controller's hardware documentation.
FDDI	*A Primer on FDDI: Fiber Distributed Data Interface* available from HP for details about the features, topologies, and components of the FDDI local area network standard.
Gateway Routing Daemon (GATED)	On the World Wide Web, use URL `http://www.gated.org/`
HYPERchannel (HYPERchannel H269 driver hardware)	*H269 (Rel. 1.2) Network Adapter Driver for DEC VAX VMS Installation Manual and User's Guide* available from Network Systems Corporation, Minneapolis, MN.
ONC RPC	*Programmer's Guide* for details about TCPware's implementation of ONC RPC.
proNET (Proteon's token ring)	*Operation and Maintenance Manual for the proNET Local Network System* available from Proteon Inc., Westborough, MA
Remote magnetic tape service (rmt)	*Maintenance Commands* section of the *SunOS Reference Manual* available from Sun Microsystems.
Setting up print queues and initiating print commands on the OpenVMS host	HP's *Guide to Maintaining a VMSSystem*. OpenVMS users can also see the *VMS DCL Dictionary*, or the *DECprint Printing Services User's Guide*.
X.25	VAX P.S.I. documentation from Hewlett-Packard.

Hewlett-Packard Documentation

For details on the OpenVMS operating systems, system services, and utilities, see the appropriate Hewlett-Packard documentation.

Appendix B TCPware Logicals

Table B-1 lists the TCPware logicals in alphabetical order:

Table B-1 TCPware Logicals

FTP_STARTUP

Defines FTP_STARTUP to point to the FTP_STARTUP.COM file.

```
$ DEFINE /SYSTEM /EXECUTIVE FTP_STARTUP SYS$MANAGER:FTP_STARTUP.COM
```

Client users can override this startup file by creating their own. Including the command DEFINE /PROCESS FTP_STARTUP in a user's LOGIN.COM file overrides any DEFINE /SYSTEM /EXEC command in the SYS$MANAGER:SYSTARTUP_V5.COM file.

NETCU_STARTUP

Defines NETCU_STARTUP to point to the NETCUSTART.COM file.

For example, you can include the following in your LOGIN.COM file:

```
ASSIGN SYS$LOGIN:NETCUSTART.COM NETCU_STARTUP
```

When you start NETCU, NETCU_STARTUP points to the specified file (SYS$LOGIN:NETCUSTART.COM for example) and processes all the commands.

Note! The system ignores all commands following an EXIT or QUIT command in the file. NETCU ignores any "commented-out" command lines in files (such as SERVICES.COM) that are used as input to NETCU. The commented-out line in the file should begin with the !, the #, or the ; character. NETCU does not execute the command line until you remove the character.

SSH_DIR

Points to the directory where SSH's master server log file is kept. Normally, this is `TCPWARE_COMMON:[TCPWARE].`

SSH2_DIR

Points to the directory where the SSH master server log file is kept. Normally, this is TCPWARE_COMMON:[TCPWARE_SSH2].

SSH_EXE

Points to the directory where SSH executables are kept. Normally, this is `TCPWARE_COMMON:[TCPWARE]`. It is defined through @TCPWARE:CNFNET SSH. The configuration procedure should write these to the common configuration file and check the values at start up and delete them at shutdown.

SSH_LOG

Points to the directory where the log files are kept. Normally, this is `TCPWARE_COMMON:[TCPWARE.LOG]`. It is defined through @TCPWARE:CNFNET SSH. The configuration procedure writes these to the common configuration file and check the values at start up and delete them at shutdown.

SSH_MAX_SESSIONS

This is set to the maximum number of concurrent SSH sessions allowed to the server system. If SSH_MAX_SESSIONS is not defined, the default is 9999. Setting SSH_MAX_SESSIONS to zero (0) will cause an error. The value must be between 1 and 9999. It is defined through @TCPWARE:CNFNET SSH. The configuration procedure should write these to the common configuration file and check the values at start up and delete them at shutdown.

TCPWARE_SSH_SFTP_SERVER_DEBUG

Enables debugging messages for the SFTP-SERVER2 image that provides service to SCP2 commands that use the SFTP protocol. When this is defined, the file SFTP-SERVER.LOG is created in the user's login directory. These files are not purged. Larger values yield more debugging information.

TCPWARE_SSH_SCP_SERVER_DEBUG

Enables debugging messages for the SCP-SERVER1 image that provides service to SCP2 commands that use the RCP over SSH2 protocol. When this is defined, the file SCP-SERVER.LOG is created in the user's login directory. These files are not purged. Larger values yield more debugging information.

SSH_TERM_MBX

Mailbox used by SSHD_MASTER to receive termination messages from SSHD daemon processes. **Do not change this logical name.** This is created by the SSHD_MASTER process.

TCPWARE_ACECLIENT_CL

Points to the shareable image activated by LOGINOUT when login is performed.

TCPWARE_ACECLIENT_DATA_DIRECTORY

Points to the directory that contains ACE/Client data files. Set by the `Enter directory where the TCPware ACE/Client data file resides:` prompt in CNFNET.

TCPWARE_ACECLIENT_ENABLE

Indicates that authentication by the TCPware ACE/Client is enabled when set to 1. Set by the `Do you want to use the TCPware ACE/CLIENT to authenticate user login?`: prompt in CNFNET.

TCPWARE_ACECLIENT_NETWORK

Indicates that authentication is performed on logins over network terminals when set to 1. For example, _NT physical devices created if using TELNET. Set by the `Do you want to authenticate user network logins?` prompt in CNFNET.

TCPWARE_ACECLIENT_PASSCODE_TIME

Defines the number of seconds allowed for the user to input the PASSCODE. Set by the `Enter the PASSCODE input timeout time`: prompt in CNFNET.

TCPWARE_ACECLIENT_REMO

Indicates that authentication is performed on logins over remote terminals when set to 1. For example, _RT physical devices are created if using SET HOST. Set by the `Do you want to authenticate user remote logins?`: prompt in CNFNET.

TCPWARE_ACECLIENT_SHR

Points to the ACE/Client API.

TCPWARE_DOMAINLIST

Allows you to set up to six domains in a search list, as well as the minimum number of dots to recognize in a host name to make it fully qualified. The client reads this information from this logical through CNFNET.

TCPWARE_DOMAINNAME

Specifies the internet addresses of up to three name servers the client can query. The client reads this information from this logical through CNFNET.

TCPWARE_FTP_220_REPLY

Defines a message displayed when a user connects to the server and can log in. This message replaces the default message.

For example, you can define the welcome text equivalence string as follows:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_220_REPLY -
_$ "**AUTHORIZED USE ONLY **",-
_$ "bart.nene.com (192.168.34.56)", -
_$ "FTP-OpenVMS FTPD V5.9 (c) 2007 Process Software"
```

Alternately, you can include the last three equivalence strings in an FTP_WELCOME.TXT file and define the logical as follows:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_220_REPLY -
_$ "@SYS$MANAGER:FTP_WELCOME.TXT"
```

In either case, when a user connects to a host, the message appears as follows:

```
220-** AUTHORIZED USE ONLY **
220-bart.nene.com (192.168.34.56)
220 FTP-OpenVMS FTPD V5.9 (c) 2007 Process Software
_Username []:
```

TCPWARE_FTP_221_REPLY

Defines a message to appear when a user ends the FTP session. If not defined, TCPware uses the default message. You can define a text string or file.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_221_REPLY -
_$ "Connection to FTP server has been closed"
```

Now, when the user closes the FTP connection, the following message appears:

```
221 Connection to FTP server has been closed
```

TCPWARE_FTP_230_REPLY

Defines a message to appear when a user successfully logs in. If not defined, TCPware uses the default message. You can define a text string or file. For example:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_230_REPLY "Login successful"
```

Now, when the user logs in using FTP, the following message appears:

```
230 Login successful
```

TCPWARE_FTP_421_REPLY

Defines a message sent when a user connects to the server but should not log in. After sending the message, the connection closes. For example, you can define this logical to prevent FTP access for a short time period. Be sure to deassign the logical after this period to allow FTP access again. You can define a text string or file.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_421_REPLY-
_$ "System maintenance in progress until 17:30"
```

Now, when the user connects to the host through FTP, the following message appears and then the connection closes:

```
421 System maintenance in progress until 17:30
```

TCPWARE_FTP_421_REPLY has precedence over TCPWARE_FTP_220_REPLY.

TCPWARE_FTP_ACCESS

TCPWARE_FTP_<username>_ACCESS

These SYSTEM logical names are used to specify the types of access that the user of the FTP server is not allowed to perform. TCPWARE_FTP_ACCESS controls all users that do not have TCPWARE_FTP_<username>_ACCESS defined. The values are:

- D - Delete
- L - List (Directory)
- R - Read
- S - Spawn
- W - Write

$ DEFINE/SYSTEM/EXECUTIVE TCPWARE_FTP_ANONYMOUS_ACCESS WDS

Will prevent the user ANONYMOUS from storing files on the system, deleting files that are present on the system or using the site specific spawn command.

TCPWARE_FTP_ADD_CC_ON_FIXED_RECORD_FILES

When the logical TCPWARE_FTP_ADD_CC_ON_FIXED_RECORD_FILES is defined to TRUE and a file is transferred as TYPE IMAGE with QUOTE SITE RMS BLOCK OFF in effect, the FTP server will separate the records of a fixed length record file with the linefeed character. This is useful for avoiding the explicit conversion necessary when transferring the file to a non-VMS system with an FTP client that is not able to do record mode transfers.

TCPWARE_FTP_ALL_VERSIONS

Requests the NLST and LIST commands to display all versions of the specified files. If TCPWARE_FTP_ALL_VERSIONS is defined, TCPWARE_FTP_STRIP_VERSION has no effect.

TCPWARE_FTP_ALL_VERSIONS is ignored if the FTP server is in UNIX emulation mode.

TCPWARE_FTP_ALLOWCAPTIVE

By default, the FTP server does not allow file transfers for CAPTIVE accounts. Defining this logical allows CAPTIVE accounts to use all FTP commands except SITE SPAWN.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_ALLOWCAPTIVE " "
```

You must modify the CAPTIVE account procedure to allow the FTP server to start the data transfer process. The procedure can check if the logical "TT" is equal to "TCPWARE:FTPSERVER_DTP.COM" and exit out of the login procedure:

```
$! Check if this is the TCPware FTP data transfer process:
$ IF F$LOGICAL("TT") .EQS. "TCPWARE:FTPSERVER_DTP.COM" THEN EXIT
$! Refuse other network connections (such as DECnet):
$ IF F$MODE() .EQS. "NETWORK" THEN LOGOUT
$! (or allow by using "...THEN EXIT" above)
$! Remainder of CAPTIVE procedure follows:
$....
```

TCPWARE_FTP_ANONYMOUS_230_REPLY

Defines a message to appear when an ANONYMOUS user successfully logs in. If not defined, TCPware uses the default message. You can define a text string or file.

```
$ DEFINE/SYSTEM/EXECUTIVE TCPWARE_FTP_ANONYMOUS_230_REPLY-
_$ "ANONYMOUS login successful"
```

Now, when a user logs in using the ANONYMOUS account, the following message appears:

```
230 ANONYMOUS login successful
```

TCPWARE_FTP_ANONYMOUS_RIGHTS

Defines write, rename, and delete access rights for the ANONYMOUS FTP user in addition to read access.

```
$ DEFINE/SYS/EXEC/NOLOG TCPWARE_FTP_ANONYMOUS_RIGHTS "WRITE,RENAME,DELETE"
```

TCPWARE_FTP_ANONYMOUS_ROOT

Defines access restrictions for users logged in as ANONYMOUS. For example, you can set access restrictions for users logged in as ANONYMOUS to allow access to just the ANONYMOUS$USER directory and its subdirectories:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_ANONYMOUS_ROOT ANONYMOUS$USER:
```

If not set, the FTP server defaults to the setting in TCPWARE_FTP_ROOT if it exists.

TCPWARE_FTP_DISALLOW_UNIX_STYLE

Controls whether UNIX style filename parsing is done. If not defined, it defaults to TRUE (UNIX-style life specifications are not allowed). Defining to FALSE allows file specifications with the “/” character in them to be treated as UNIX file specification.

```
$ DEFINE /SYSTEM/ NOLOG /EXECUTIVE TCPWARE_FTP_DISALLOW_UNIX_STYLE ?
```

TCPWARE_FTP_DONT_REPORT_FILESIZE

If this logical is defined, the reporting of the estimate of the number of bytes to be transferred in the 150 response line is suppressed. Some FTP clients expect this number to be exact. The FTP server is unable to determine an exact count without processing the entire file, so an estimate of the number of bytes used to store the file is returned. The inaccuracy comes from the differences in the way OpenVMS records and line breaks are handled. The ? in the logical represents where defined values go.

```
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_DONT_REPORT_FILESIZE ?
```

TCPWARE_FTP_EXTENSION_QUANTITY

Defines the default allocation/extension quantity for new files and appends. The ? in the logical represents where defined values go. Defined values must be numeric.

```
$ DEFINE /SYSTEM/ NOLOG /EXECUTIVE TCPWARE_FTP_EXTENSION_QUANTITY ?
```

TCPWARE_FTP_GETHOST_MAX_TIME

When a new connection arrives at the FTP server it attempts to resolve the name of the host that originated the connection. If this process takes a long time, it can stall all other connections, both active and new. To adjust how long the FTP server is allowed to take to look up the host name, set the logical TCPWARE_FTP_GETHOST_MAX_TIME to the VMS delta time that can elapse before it gives up. The default value 10 seconds (0 0:0:10).

TCPWARE_FTP_IDLE_TIMEOUT

Changes the timeout for FTP connection attempts to something other than the default of 10 minutes. The FTP server checks the timeout when you enter and complete a command. You can set this logical any time, and it effectively changes the idle timeout for open, non-idling connections as well as for any future ones. Make sure to use delta time for the time syntax.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_IDLE_TIMEOUT "0 00:20:00"
```

This example changes the idle timeout to 20 minutes. The default is 10 minutes if no time is specified. Setting the value to 0 disables idle timeout.

TCPWARE_FTP_IGNORE_UNIX_DASH_OPTIONS

By default, the FTP server ignores Unix-style dash options on LIST and NLST when in Unix mode (for example, the “-l” in “ls -l”). Define this to be FALSE to tell the FTP server to pay attention to Unix-style dash options.

```
$ DEFINE /SYSTEM /EXEC TCPWARE_FTP_IGNORE_UNIX_DASH_OPTIONS FALSE
```

TCPWARE_FTP_KEEP_DIR_EXT

Sometimes the FTP server strips the .DIR extension from the file name of a directory when the NLST function is requested. The FTP server looks for TCPWARE_FTPD_KEEP_DIR_EXT and, if defined, does not remove the .DIR extension.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTPD_KEEP_DIR_EXT TRUE
```

To return to the default behavior, remove this logical.

TCPWARE_FTP_MAXIMUM_CONNECTION_WAIT

A VMS delta time for how long the FTP client (or programming library) should wait for the 220 response after connecting to the FTP server.

TCPWARE_FTP_NOKEEPALIVES

When TCPWARE_FTP_NOKEEPALIVES is defined, the FTP server will not send keepalives on the control channel. The KEEPALIVE command allows the FTP client program to toggle, regardless of whether or not it desires keepalives to be sent on the control channel. The SET [NO]KEEPALIVE command allows the FTP client to explicitly set whether or not it desires keepalives on the control channel

TCPWARE_FTP_LOGFILE

Defines a specific name of a log file. Use this if you suspect break-ins to the FTP server.

```
$ DEFINE /SYSTEM /EXEC TCPWARE_FTP_LOGFILE SYS$COMMON:[SYSMGR]FTPLOGIN.LOG
```

This logical must be defined before TCPware FTP is started (or FTP must be restarted after defining it in order for it to take effect).

If this logical exists, the FTP server writes a record to the specified file each time a user attempts to log in. Each record includes the date and time, the remote host's internet address, and whether the login succeeded.

Specifies the name of the file to which ALL commands and responses to ANONYMOUS FTP services are logged. If TCPWARE_FTP_LOG_ALL_USERS is also defined, then commands and responses for all users are logged.

TCPWARE_FTP_LOG_ALL_USERS

This logical causes all commands and responses to be logged to the file defined by TCPWARE_FTP_LOGFILE. The default (when this logical is not defined) is to just log the commands and responses for anonymous users.

```
$ DEFINE TCPWARE_FTP_LOG_ALL_USERS
```

TCPWARE_FTP_MAX_SERVERS

Allows the maximum number of servers to be set. The default is 10000.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_MAX_SERVERS "1500"
```

TCPWARE_FTP_MAXREC

The FTP client and the FTP server check the record size of an ASCII transfer and disallow more than 8192 byte records. Define this logical to override the default of 8192. The definition of this logical is commented out but defined in the FTP_CONTROL.COM file as follows:

```
$ !DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_MAXREC 8192
```

TCPWARE_FTP_MESSAGE_FILE

Defines the message file the FTP user sees when connecting to the server or moving between directories. The definition of this logical is commented out but defined in the FTP_CONTROL.COM file as follows:

```
$ !DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_MESSAGE_FILE ".MESSAGE"
```

TCPWARE_FTP_ONLY_BREAK_ON_CRLF

If this logical is set and an ASCII file is transferred, a new line is created in the file upon receipt of a carriage return/line feed sequence.

If this logical is not set and an ASCII file is transferred, a new line is created upon receipt of either a carriage return/line feed sequence or a line feed.

TCPWARE_FTP_PASSWORD_WARNING_MESSAGE

The logical TCPWARE_FTP_PASSWORD_WARNING_MESSAGE defines the message that the FTP server displays when the user's password is going to expire within the warning time. If the amount of time before the password expires is to be displayed, use a %s in the logical.

```
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_PASSWORD_WARNING_MESSAGE "%s"
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_PASSWORD_WARNING_MESSAGE "message text
string"
```

TCPWARE_FTP_PASSWORD_WARNING_TIME

The logical TCPWARE_FTP_PASSWORD_WARNING_TIME uses the VMS delta time to specify the minimum remaining lifetime for the user's password. If the remaining lifetime is greater than the VMS delta time then no message is displayed. It is necessary to define this value to enable checking for the remaining lifetime of a password.

```
$ DEFINE/SYSTEM/EXEC @TCPWARE_FTP_PASSWORD_WARNING_TIME “dddd hh:mm:ss.hh”
```

TCPWARE_FTP_RECEIVE_THRESHOLD

Specifies the amount of buffer space that can be used to buffer transmitted data on the data socket. The default value if 6144. If this logical is defined and it begins with a /, then it specifies the fraction of the window size; if only a fraction is specified, then it indicates the number of bytes to be used. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE TCPWARE_FTP_RECEIVE_THRESHOLD ?
```

TCPWARE_FTP_RECODE_NONVMS_FILE_NAMES

If this logical is defined, and the FTP server is not operating in UNIX mode, it recodes filenames that are not legal OpenVMS file names in the same manner that it would normally recode filenames when operating in UNIX mode. This is useful for handling filenames with multiple dots (.), spaces, and other characters that VMS does not allow in filenames while retaining the OpenVMS directory syntax.

```
$ DEFINE TCPWARE_FTP_RECODE_NONVMS_FILE_NAMES filename
```

TCPWARE_FTP_ROOT

Defines the system-wide default directory access restrictions for client users. The logical may be defined as a single directory or a search list of directories.

For example, you can restrict all users logged in via FTP to the COMMON$USER directory and its subdirectories:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_FTP_ROOT COMMON$USER:
```

The default directory is not set to the value of this logical or to the value of TCPWARE_FTP_<*username*>_ROOT.

TCPWARE_FTP_*username*_ROOT

The TCPWARE_FTP_*username*_ROOT (system level, executive mode) logical defines access restrictions for an FTP client logging in as *username*. The logical may be defined as a single directory or a search list of directories.

For example, you can restrict user CLARK to the COMMON$USER:[CLARK] directory and its subdirectories, as follows:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_CLARK_ROOT COMMON$USER:[CLARK]
```

Because the FTP server restricts access by default to the directory setting in the TCPWARE_FTP_ROOT logical (described earlier), if it exists, you may want to use the special wildcard (*) setting with the TCPWARE_FTP_*username*_ROOT logical to bypass the default for *username*. For example, to restrict the bulk of users to DISK$SYS_LOGIN, restrict users KATE and PAUL to ENG$DISK, but allow SYSTEM full access to locations covered by its account, define the following logicals:

```
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_ROOT DISK$SYS_LOGIN ! default
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_KATE_ROOT ENG$DISK  ! limits KATE
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_PAUL_ROOT ENG$DISK  ! limits PAUL
$ DEFINE/SYSTEM/EXEC TCPWARE_FTP_SYSTEM_ROOT *       ! full SYSTEM
```

ANONYMOUS user access restrictions are described under *TCPWARE_FTP_ANONYMOUS_ROOT*.

The user is not placed automatically in this directory upon successful login.

TCPWARE_FTP_SEMANTICS_FIXED_IGNORE_CC

If this logical is defined to TRUE, then GET operations of fixed lengths record files will not have a carriage return/line feed added to the end of each record. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE TCPWARE_FTP_SEMANTICS_FIXED_IGNORE_CC ?
```

TCPWARE_FTP_SEMANTICS_VARIABLE_IGNORE_CC

When this logical is defined to TRUE, files with variable length records and carriage return carriage control will NOT have a new line character inserted after each line when the file is transferred in image (binary) mode. The default is TRUE and is defined in FTPSERVER_DTP.COM.

```
$ DEFINE TCPWARE_FTP_SEMANTICS_VARIABLE_IGNORE_CC FALSE
```

Users can change this value by defining it in their LOGIN.COM file, or it can be defined on a system-wide basis if this is desired for all users.

TCPWARE_FTP_SEND_FEAT_ON_CONNECT

By default, the FTP client sends the FEAT command upon connecting to a server. This can be disabled by defining this logical as FALSE.

```
$ DEFINE TCPWARE_FTP_SEND_FEAT_ON_CONNECT FALSE
```

When this is disabled the FTP client will not be able to detect the support of optional features such as TLS, REST STREAM, and others and these features may not work correctly if there is an attempt to use them.

TCPWARE_FTP_SERVER_DATA_PORT_RANGE

Specifies the upper and lower port boundaries that are to be used in passive data connections. The string should contain two numbers separated by a space. The ? in the logical represents where defined values go.

```
$ DEFINE TCPWARE_FTP_SERVER_DATA_PORT_RANGE ?
```

TCPWARE_FTP_SERVER_LOG_LIMIT

By setting this logical in the LOGIN.COM file, you can specify that log files be retained. Set the logical name to a dash (-) to retain all log files, or specify a number in the range of 1 to 32000.

Directory size restrictions limit the number of potential files that can be created. If you do not specify a number or value, one log file is created or overwritten for each FTP session. Use the DCL PURGE command to delete unneeded log files. The following example specifies that 42 log files be retained:

```
$ DEFINE TCPWARE_FTP_SERVER_LOG_LIMIT 42
```

TCPWARE_FTP_SERVER_RELAXED_PORT_COMMAND

The server compares the IP network address value specified in the PORT command with the IP network address of the IP address it is receiving commands from. If these are not in agreement, the PORT command is not accepted. Some multi-homed clients, and clients that can do third party transfers, send values that do not match. Defining this logical allows the PORT command to be accepted for these clients by disabling this check. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE TCPWARE_FTP_SERVER_RELAXED_PORT_COMMAND ?
```

TCPWARE_FTP_STRIP_VERSION

Causes VMS mode output to have no versions. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_STRIP_VERSION ?
```

TCPWARE_FTP_SYST_BANNER

When this logical is defined the SYSTem banner is not displayed in response to the STATUS command. When this logical is not defined the format of the banner varies depending upon whether the FTP_SERVER is operating in UNIX mode or VMS mode.

```
$ DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_SYST_BANNER
```

TCPWARE_FTP_UNIX_STYLE_BY_DEFAULT

Starts the FTP server in UNIX emulation mode. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_UNIX_STYLE_BY_DEFAULT ?
```

When sending the command from a non-OpenVMS client, a space is required between the file specification and the qualifier.

```
$ GET filename /LOG
```

To disable this requirement:

```
$ DEFINE /SYSTEM /EXECUTIVE_MODE TCPWARE_FTPD_NOUNIX_SYNTAX “TRUE”
```

This logical has no effect if TCPWARE_FTP_DISALLOW_UNIX_STYLE is not set to FALSE.

TCPWARE_FTP_UNIX_STYLE_CASE_INSENSITIVE

Allows UNIX style filename handling to be case insensitive. The ? in the logical represents where defined values go. Defined values can be either alpha or numeric.

```
$ DEFINE /SYSTEM /NOLOG /EXEC TCPWARE_FTP_UNIX_STYLE_CASE_INSENSITIVE ?
```

TCPWARE_FTP_USE_SRI_ENCODING_ON_ODS5

This logical can be defined to 1, TRUE or YES to cause the filename encoding used for UNIX-style filenames on ODS-2 disks to be used on ODS-5 disks. This also sets the default case of letters in filenames to lowercase and ignores the stored case.

TCPWARE_FTP_WINDOW

The FTP client and the FTP server set the TCP window size of the data connection to either:

- The value of this logical if you define it (minimum is 512 bytes; maximum is 1,048,576 bytes)
- The larger of 32,768 bytes and the default TCP window size

The ? in the logical represents where defined values go. Defined value should be numeric.

```
$ DEFINE /SYSTEM /NOLOG /EXECUTIVE TCPWARE_FTP_WINDOW ?
```

TCPWARE_IMAP_UPDATE_LOGIN_TIME

If this logical is defined (to any value), then IMAP updates the user's "Last login: (non-interactive)" field on the server with the last time the user downloaded his/her mail via an IMAP client.

TCPWARE_KERBV4_MAXAGE

Sets the maximum age of the Kerberos database.

TCPWARE_KERBV4_PRIMARY

Sets the primary Kerberos server name.

TCPWARE_KERBV4_REALM

Sets the realm name of the Kerberos server.

TCPWARE_KERBV4_RLOGIN

Determines if the RLOGIN server mandates, accepts, or disallows any Kerberos request.

TCPWARE_KERBV4_RSHELL

Determines if the RSH server mandates, accepts, or disallows any Kerberos request.

TCPWARE_KERBV4_SRVTYP

Sets the type of server (primary or applications only).

TCPWARE_KERBV4_TELNET

Determines if the TELNET server mandates, accepts, or disallows any Kerberos request.

TCPWARE_KERBV4_TKFILE

Sets the location of the user's ticket file.

TCPWARE_LPD_DEFAULT_USER

Defines a default OpenVMS username for remote users connecting to the local LPD server. Used only when you define a remote host in the LPD access file and the remote username is not mapped to a specific OpenVMS username.

TCPWARE_LPD_OPTIONS

Determines if the server handles batch queues.

TCPWARE_LPD_*qname*_*_FORM

Defines the form used for print jobs. This is similar to TCPWARE_LPD_*qname*_PARAMETER.

Use TCPWARE_LPD_*_FORM to define the form for all queues.

Note! A specific queue setting overrides the global setting for that queue.

TCPWARE_LPD_*qname*_OPTION

Specifies additional PRINT command qualifiers to pass to the specified print queue:

/BURST, /FEED, /FLAG, /FORM, /HEADER, /LOWERCASE, /PASSALL, /PRIORITY, /RESTART, /SPACE, /TRAILER

Use TCPWARE_LPD_*_OPTION to define the option for all queues.

Note! A specific queue setting overrides the global setting for that queue.

TCPWARE_LPD_*qname*_*_PARAMETER

Defines the specified parameters when the remote user submits a print request to the OpenVMS print system (*qname* is the queue name).

The first equivalence string for the logical (if defined) is the first parameter; the second is the second parameter; and so on, up to eight parameters.

Use TCPWARE_LPD_*_PARAMETER to define the parameter for all queues.

Note! A specific queue setting overrides the global setting for that queue.

TCPWARE_LPD_*qname*_*_QUEUE

Defines the print queues for an alias queue name (*qname*). Supports clients that may not allow standard OpenVMS queue names as the remote printer (such as IBM's AIX, which restricts remote printer names to seven characters).

TCPWARE_LPD_SPOOL

Points to the work directory for the LPD server. This directory holds temporary files.

TCPWARE_LPR_PRINTER

Defines the default remote printer for the LPR, LPRM, and LPQ commands. Define your own TCPWARE_LPR_PRINTER logical in a LOGIN.COM file.

TCPWARE_LPR_*qname*_PRINTER
TCPWARE_LPR_*qname*_PRINTER_DEFAULT

Defines the absolute printer for the PRINT command. You cannot override this logical when submitting a print job. Use to restrict printing to one printer per queue.

TCPWARE_LPR_QUEUES

Lists the names of all TCPware print symbiont queues. Defined only if you defined one or more print queues.

TCPWARE_LPR_SPOOL

Points to the work directory for the PRINT command. This directory holds temporary files.

TCPWARE_LPRSM

The TCPWARE_LPRSMB print symbiont provides similar retry interval and timeout tuning logicals as those for TCPWARE_VMSLPRSMB. The TCPWARE_LPRSMB logicals are:

- TCPWARE_LPRSMB_*_RETRY_INTERVAL
- TCPWARE_LPRSMB_*qname*_RETRY_INTERVAL
- TCPWARE_LPRSMB_*_TIMEOUT
- TCPWARE_LPRSMB_*qname*_TIMEOUT
- TCPWARE_LPRSM_*qname*_PRECONN

TCPWARE_NAMED_MAX_CACHE_TTL

NAMED checks the SYSTEM EXECUTIVE logical table for this logical value and sets the maximum cache time (in seconds) to be that value. Use this logical to override the default one week (604800 seconds) to a maximum cache time more appropriate for your system.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_NAMED_MAX_CACHE_TTL 86400
```

The server reads this logical the next time it starts. If you do not want to wait for the server to start, you can make the change to the running server by using the NETCU SET NAMED MAX_TTL command. Any data now written to the cache remains there for 86400 seconds (one day).

TCPWARE_NAMESERVERS

When an application needs to resolve a host name or internet address, the client queries the first name server this logical defines. The client continues to query the other name servers on its list until it receives an answer or the list is exhausted.

TCPWARE_NFS_ACCESS_IDENTIFIER

Specifies the name of a rights identifier you want assigned to all NFS users. You can then modify the access control lists (ACLs) of files to grant or deny access to holders of the rights identifier. The default is null (no rights identifier).

OpenVMS files protected by ACLs should have the UIC-based protection mask set to allow file access and the ACL set to deny access.

TCPWARE_NFS_DFLT_GID
TCPWARE_NFS_DFLT_UID

Specifies the default UID and GID. The server uses these defaults in the following cases:

- Receives a request from a user without a PROXY mapping and who is also the superuser (UID=0, and any GID). The server replaces the superuser UID and GID with the default UID and GID.
- Processes a `get attributes` request and cannot find a file's owner UIC in the PROXY database. The server uses the default UID and GID instead.

TCPWARE_NFS_DIRLIFE_TIMER

Sets when to delete internal directory cache data structures. Specify the interval as OpenVMS delta time. The default is 3 minutes.

TCPWARE_NFS_DIRREAD_LIMIT

Sets the maximum size in bytes for each file read while processing a `get attributes` request. If the estimated file size exceeds this value, TCPware does not read the file to determine its exact size and returns an estimated size instead. The estimated file size is always larger than the exact size. The `-1` default turns off file size estimation.

This parameter applies only to filesystems exported with the /CONVERT option (the default). A value of `0` disables TCPware from determining exact file sizes on requests.

This parameter may provide the NFS Client with inexact file sizes. This is not a problem, but may affect some applications.

TCPWARE_NFS_DIRTIME_TIMER

Sets a time interval that determines when the server updates the directory access time between NFS operations. Specify the interval as an OpenVMS delta time. The default is 30 seconds.

TCPWARE_NFS_DYNAMIC_EXPORT

Reloads updates to the shared database on the cluster automatically when you set this logical to CLUSTER:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_NFS_DYNAMIC_EXPORT CLUSTER
```

The server uses locks to communicate changes to all the servers on the cluster. The default is LOCAL (not to use locks).

TCPWARE_NFS_DYNAMIC_PROXY

Enables dynamic PROXY database reloading.

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_NFS_DYNAMIC_PROXY keyword[,keyword]
```

The *keywords* are

- CLIENT -- enables Client reloading
- SERVER -- enables Server reloading
- NOCLIENT and NOSERVER -- when used with the ADD PROXY or REMOVE PROXY commands overrides the logical setting

TCPWARE_NFS_FILE_CACHE_SIZE

Determines the maximum number of files allowed to have attributes in cache at any one time. The number must be larger than the SYSGEN parameter CHANNELCNT. The value must also be larger than the number of combined TCP and UDP threads.

TCPWARE_NFS_LOG_CLASS

Enables the type of information written to the log file TCPWARE:NFSSERVER.LOG. This parameter is a bit mask value (in decimal).

TCPWARE_NFS_NOCHECKSUM

Enables or disables checksum generation for UDP datagrams. This parameter is a boolean value. When the value is 0 (false), the server generates checksums for outgoing datagrams. When the value is 1 (true), the server does not generate checksums. Enabling checksums maintains data integrity, and is the default.

Note! Disabling checksums may increase system performance but could have an adverse affect on certain NFS clients.

TCPWARE_NFS_OPENFILE_TIMER

Sets a time interval (in delta time) a file remains open after you last accessed it. You do not need to open and close it for each request. The default is six seconds.

TCPWARE_NFS_PCNFSD_DFLTPRTOPT

Specifies the default print options when submitting a spooled print job for printing. The logical for NFS_PCNFSD_DFLPRTOPT is TCPWARE_PCNFSD_DFLTPRTOPT.

TCPWARE_NFS_PCNFSD_ENABLE

Enables (value of 1) or disables (value of 0) the PCNFSD services support. A value of 3 enables print spooling of files on the server without enabling PCNFSD authentication. The logical for NFS_PCNFSD_ENABLE is TCPWARE_PCNFSD_ENABLE.

TCPWARE_NFS_PCNFSD_JOB_LIMIT

Specifies the maximum packet size of the information displaying the queued print jobs. Some systems require this limitation.

Note! If the actual queued job information exceeds the byte limit set by this parameter, TCPware truncates the information.

The logical name for NFS_PCNFSD_JOB_LIMIT is TCPWARE_PCNFSD_JOB_LIMIT. If this logical is not defined, TCPware determines the size of the packet at run-time.

TCPWARE_NFS_PCNFSD_PRINTER

Specifies the print queue you want used if the NFS client does not specify a printer. This is an optional parameter and the default is SYS$PRINT when the client does not specify a printer (most clients specify the printer). The logical for NFS_PCNFSD_PRINTER is TCPWARE_PCNFSD_PRINTER.

TCPWARE_NFS_PCNFSD_PRINTER_LIMIT

Specifies the maximum packet size of the information displaying the printers known on the server. Some systems require this limitation.

Note! If the actual printer information exceeds the byte limit set by this parameter, TCPware truncates the information.

The logical for NFS_PCNFSD_PRINTER_LIMIT is TCPWARE_PCNFSD_PRINTER_LIMIT. If this logical is not defined, TCPware determines the size of the packet at run-time.

TCPWARE_NFS_PCNFSD_SPOOL

Specifies the name of the PCNFSD print spool directory as a UNIX style pathname. The directory must be an exported directory. This is, the directory must be an entry in the EXPORT database, or a subdirectory of an exported directory. The logical for NFS_PCNFSD_SPOOL is TCPWARE_PCNFSD_SPOOL.

Because you export different OpenVMS directories to different clients with the same path, it is possible for the NFS_PCNFSD_SPOOL parameter to refer to different OpenVMS directories depending on which PCNFSD client requests the print spooling services.

TCPWARE_NFS_PORT

Sets the TCP and UDP port through which the NFS, MOUNT, and PCNFSD protocols receive data.

TCPWARE_NFS_SECURITY

Enables various security features. This parameter is a decimal bit mask value.

Caution! Do not use bits 0 and 1 for PC clients using PCNFS.

If you use PC-NFS printing with mask value=2, add an entry to the EXPORT database for each client subdirectory (not just a single entry for the spool directory.) The pathname listed in the EXPORT database should be the NFS_PCNFSD_SPOOL parameter value concatenated with the name of the client subdirectory.

If you set bit 5, PC-NFS users can print to batch queues. This may present a security risk, since users could submit batch jobs under a privileged (or another) user by forcing the UID/GID values of their choice.

Disabling use of the intrusion database for PCNFSD, by setting bit 6, affects all exports.

A bit mask 8 value of 128 disables PCNFSD deletion of printed files from the spool directory.

TCPWARE_NFS_TCP_THREADS

Controls the number of simultaneously serviced requests received over TCP connections the server can support. The server requires a thread for each TCP request it receives. This thread is active for the amount of time it takes the server to receive the request, perform the operation, and send a reply to the client.

The more threads the server supports, the better the performance.

Note! The number of threads has no impact on the number of TCP connections the server supports.

TCPWARE_NFS_UDP_THREADS

This is similar to the NFS_TCP_THREADS parameter but relates to UDP threads.

TCPWARE_NFS_XID_CACHE_SIZE

Sets the maximum number of XID cache entries. The XID cache prevents the system from transmitting false error messages for operations such as delete, create, rename, and set attributes.

Set the NFS_XID_CACHE_SIZE parameter to at least twice (2 times) the largest of the number of:

- NFS clients using the NFS Server
- UDP threads (as set by the NFS_UDP_THREADS parameter)
- TCP threads (as set by the NFS_TCP_THREADS parameter)

The parameter sets the size of both the UDP and TCP XID caches (each protocol has a separate XID cache).

TCPWARE_PCNFSD_DFLTPRTOPT

Specifies the default print options when submitting a spooled print job for printing. The logical for NFS_PCNFSD_DFLPRTOPT is TCPWARE_PCNFSD_DFLTPRTOPT.

TCPWARE_POP3_UPDATE_LOGIN_TIME

If this logical is defined (to any value), then POP3 updates the user's "Last login: (non-interactive)" field on the server with the last time the user downloaded his/her mail via an POP3 client.

TCPWARE_PPPD_DEBUG_LEVEL

When you specify the DEBUG (or -D) option, it debugs at level 5 (display up to warning and significant events). For more informational and debugging information, raise the debug level to 7.

TCPWARE_PPPD_OPCOM_LEVEL

For a detached process, raise the message level for OPCOM messages. By default, it is set to 4 to report fatal and error messages. Raise it to 5 to monitor the significant events in PPPD, or even higher for more detail.

TCPWARE_QUOTE

Defines the quote for the server. This logical can be either a string or a filename that includes the quote text. Prefix a filename with the `@` sign and enclose the definition or filename in quotation marks.

You need SYSNAM or SYSPRV privileges to define the system-wide logical.

```
$ DEFINE/SYSTEM/EXECUTIVE TCPWARE_QUOTE "Quote-of-the-day"
$ DEFINE/SYSTEM/EXECUTIVE TCPWARE_QUOTE "@SYS$MANAGER:QUOTE.TXT"
$ DEFINE/SYSTEM/EXECUTIVE TCPWARE_QUOTE "Today's quote is",-
_$ "@SYS$MANAGER:QUOTE.TXT"
```

TCPWARE_RCMD_FLAGS

Set this logical to `1` (default = `0`) to disable user-specified `SYS$LOGIN:.RHOSTS` files (and use the `HOSTS.EQUIV` file only).

TCPWARE_RCMD_OUTPUT

Sets up a log file for incoming R Services such as RCP and RSH to log messages in the RCMD.LOG file:

```
$ DEFINE /SYSTEM /EXECUTIVE TCPWARE_RCMD_OUTPUT RCMD.LOG
```

TCPWARE_RES_OPTIONS *ndots ndots*

Sets up to six domains in a search list, as well as the minimum number of dots to recognize in a host name to make it fully qualified. The client reads this information from two logicals you set through CNFNET.

TCPWARE_RES_RETRANS_MIN

Specifies minimum retransmit time value in seconds.

TCPWARE_RES_RETRIES

Specifies retry count.

TCPWARE_SCP_VMS_MODE_BY_DEFAULT True|Yes|1

When this logical is defined to True, Yes, or 1, the SCP command defaults to /VMS if neither /NOVMS nor /TRANSLATE_VMS are specified.

TCPWARE_SCP2_CONNECT_TIMEOUT

This logical defines a number specifying how long SCP2 should wait for a response to the INITIALIZE command from the server program. This is a VMS delta time number. The default is 2 minutes.

TCPWARE_SCP2_VMS_MODE_BY_DEFAULT

When defined to TRUE, YES, or 1, this logical chooses the /VMS qualifier if /TRANSLATE_VMS or /NOVMS has not been specified.

TCPWARE_SFTP_CASE_INSENSITIVE

This logical causes SFTP to treat filenames in a case insensitive manner when it is defined to TRUE, YES, or 1.

TCPWARE_SFTP_FALLBACK_TO_CBT True|Yes|1

When this logical is defined to TRUE, YES, or 1, and files are being transferred in VMS mode, a contiguous file will be created as contiguous best try if there is insufficient space to create it as contiguous.

TCPWARE_SFTP_FILE_ESTIMATE_THRESHOLD

This logical controls the minimum number of blocks that a text file must be for an estimated transfer size to be returned instead of an exact size. The default is to estimate the transfer size for all text files.

TCPWARE_SFTP_DEFAULT_FILE_TYPE_REGULAR

If this logical is defined to TRUE, YES or 1, then the SFTP server will use a default file type of REGULAR instead of UNKNOWN for OPEN operations. This can correct problems with filenames without a . (dot) in them getting .dir added to them. The filename will appear with a . at the end of the name in directory listings.

TCPWARE_SFTP_MAXIMUM_PROTOCOL_VERSION

This logical can be used to limit the version of the SSH File Transfer Protocol that the SFTP client and Server use. This can sometimes provide a work-around for problems encountered with different implementations of the protocol. The default value is 4. Protocol versions 2 and 3 are also used by popular implementations.

TCPWARE_SFTP_NEWLINE_STYLE

This logical controls the newline style that SFTP uses. Which can be helpful in transferring text files. The values are: UNIX <lf>, VMS <lf>, MAC <cr>. If the logical is not defined, or defined to any other value, then <cr><lf> will be used for the text line separator as documented in the SSH File Transfer specification.

TCPWARE_SFTP_ODS2_SRI_ENCODING

This logical controls whether or not SRI encoding is used for filenames on VMS ODS-2 disks. If the logical is not defined, or is defined to TRUE, YES, or 1 then SRI encoding is used on ODS-2 disks for filenames that contain uppercase letters and special characters.

TCPWARE_SFTP_RETURN_ALQ True|Yes|1

When defined to TRUE, YES or 1 and files are being transferred in VMS mode, this logical causes the allocation quantity to be transmitted when a file is transferred. Normally this value is only sent when necessary to avoid having an excessive amount of space allocated to a file when it is transferred from a disk with a large allocation cluster to a disk with a small allocation cluster.

TCPWARE_SFTP_TRANSLATE_VMS_FILE_TYPES number

When this logical is defined, the SFTP server will translate text files to stream linefeed format so that they are compatible with UNIX systems. The number is a bit mask, with the following definitions:

bit 0 (value 1) FIXED format files should be translated

bit 1 (value 2) VARIABLE format files should be translated

bit 2 (value 4) VARIABLE, FIXED CONTROL (VFC) files should be translated.

These values can be added together to specify combinations of file types. Due to the way the SCP2 client is implemented, this logical also serves as a default for the SCP2 client.

The SCP-SERVER1 program always translates FIXED, VARIABLE and VFC files as it is designed to service requests that come from UNIX systems that use the OpenSSH implementation.

TCPWARE_SFTP_VMS_ALL_VERSIONS

This logical controls whether or not all versions of a file are returned. The values TRUE, YES or 1 will return all versions, any other value is to only return the name of the file without a version. The default is to return only one filename without the version number.

TCPWARE_SLIP_*n*

The START/IP command *line-specific-information* parameter provides the OpenVMS device name for the SLIP line. If you omit this parameter, TCPware assumes that the TCPWARE_SLIP_*n* system logical (where *n* is the controller number) defines the device.

TCPWARE_LOCALDOMAIN

Specifies the default local domain name to be used when building To: addresses on outgoing messages.

For example, to have messages sent to SMTP%“Joe@construction” to be delivered to SMTP%“Joe@construction.bedrock.com”, TCPWARE_LOCALDOMAIN would be defined as “bedrock.com”.

TCPWARE_NAMESERVERS

List of IP addresses for DNS lookups.

TCPWARE_SMTP_A1_NAME

Used in forming the username portion of return addresses for ALL-IN-1 users.

TCPWARE_SMTP_ACCEPT_UNIX_LF

Tells the SMTP agents to accept lines sent by some UNIX systems that are terminated with a linefeed only (instead of the proper carriage-return, linefeed combination).

TCPWARE_SMTP_ALLOW_USER_FROM

Allows users to override their From: address on outgoing mail by specifying /FROM=xxx@yyy as the first line of outgoing mail messages.

TCPWARE_SMTP_ALLOW_VIRTUAL_DOMAIN

Allows the use of virtual domains in TCPware SMTP environment. Without this logical defined, incoming aliases are assumed to be local addresses only. If your system supports multiple virtual domains and uses in the alias file to reroute traffic based on those domains, you must define this logical.

TCPWARE_SMTP_AM_DOMAIN

Domain name used when forming return addresses for ALL-IN-1 users.

TCPWARE_SMTP_AM_NAME

Used in forming the username portion of return addresses for ALL-IN-1 users.

TCPWARE_SMTP_APPEND_FORWARDER_TO_MX

Specifies that the default SMTP forwarder, if defined, is appended to the end of an MX list for a target host when delivering outgoing mail.

TCPWARE_SMTP_BATCH_QUEUE

Points to the TCPware SMTP queue.

TCPWARE_SMTP_DECNET_DOMAIN

Specifies a DECnet name used in the creation of return addresses.

TCPWARE_SMTP_DELIVERY_RECEIPTS

Enables or disables delivery receipts (value is TRUE or FALSE).

TCPWARE_SMTP_DISABLE_DELIVERY_RECEIPT_DISCLAIMER

When deliver receipts are enabled, a disclaimer is included in all such receipts telling the sender that the message has been delivered, but not necessarily read. Defining this logical prevents the disclaimer from being included.

TCPWARE_SMTP_DISABLE_FOLDER_DELIVERY

Disables TCPware SMTP's ability to deliver messages to user-defined folders in their VMS Mail files.

TCPWARE_SMTP_DISABLE_PSIMAIL

If defined, causes mail sent to PSI% users to be returned with NOSUCHUSER.

TCPWARE_SMTP_ENVELOPE_FROM_HOST

Specifies the host name to be used in the SMTP envelope MAIL FROM: line. If not defined, the default system host name is used.

TCPWARE_SMTP_FORWARDER

Specifies the domain name of the system to which all outgoing mail is forwarded for further delivery.

TCPWARE_SMTP_FROM_HOST

Specifies the local host name used when forming From: address on outgoing messages. If this logical is not defined, the system host name is used.

TCPWARE_SMTP_HEADER_ORG

Specifies the text for an Organization: header in outgoing mail.

TCPWARE_SMTP_HEADER_RETURN_RECEIPT_TO

Generates a Return-Receipt-To: header in outgoing mail. Requires the TCPWARE_SMTP_RETURN_RECEIPT_TO_HEADER_ENABLE logical to be defined.

TCPWARE_SMTP_HEADER_SYS

Specifies the text for a System: header in outgoing mail.

TCPWARE_SMTP_HOST_ALIAS_FILE

Points to the file containing a list of all the host names that should be considered local for this node for incoming mail delivery.

TCPWARE_SMTP_HOST_NAME

Specifies all the local host names for this node. Used to specify all virtual domains handled by this node. Alternatively, the node names can be stored in the file TCPWARE:SMTP_HOST_ALIASES.

TCPWARE_SMTP_LOG

Specifies the output filename. If not defined, the name defaults to TCPWARE:TCPWARE_SMTP_LOG.queuename.

TCPWARE_SMTP_MAXIMUM_822_TO_LENGTH

Sets the maximum length of the RFC822 To: header line when delivering incoming mail to VMS Mail users.

TCPWARE_SMTP_MRGATE_NAME

Specifies the name of the Message Router gateway.

TCPWARE_SMTP_NON_LOCAL_FORWARDER

Specifies the name of a forwarder system for non-local outgoing mail.

TCPWARE_SMTP_NO_USER_REPLY_TO

Disallows the use of user-defined Reply-To: headers in outgoing mail.

TCPWARE_SMTP_POSTMASTER

Specifies the address of the system-wide postmaster.

TCPWARE_SMTP_REJECT_INVALID_DOMAINS

Tells the SMTP server to reject mail from domains whose names and addresses cannot be resolved in a reverse lookup.

TCPWARE_SMTP_REPLY_TO

Specifies an address for a Reply-To: header in outgoing mail.

TCPWARE_SMTP_RESENT_HEADERS

Causes the inclusion of "Resent-*" headers in mail forwarded from a VMS Mail account using SET FORWARD in VMS Mail.

TCPWARE_SMTP_RETRY_INTERVAL

Specifies the retry interval for messages waiting for an attempted redelivery. The time is specified as a delta time.

TCPWARE_SMTP_RETURN_INTERVAL

Specifies the amount of time a given message delivery should be retried before giving up and bouncing the message back to the sender. The time is specified as a delta time.

TCPWARE_SMTP_RETURN_MSG

Specifies an input filename for the return message SMTP sends when a mail message bounces.

TCPWARE_SMTP_RETURN_RECEIPT_TO_HEADER_ENABLE

Enables the Return-Receipt-To: header if the TCPWARE_SMTP_HEADER_RETURN_ RECEIPT_TO logical is also defined.

TCPWARE_SMTP_SEND_CLASS

Specifies the VMS broadcast class for "New mail" notifications. The default is USER16.

TCPWARE_SMTP_SERVER_DISABLE_VRFYEXPN

Disables the VRFY and EXPN commands in bitmask format to the SMTP server.
Bit 0 = VRFY; Bit 1 = EXPN.

TCPWARE_SMTP_SERVER_LOG

Enables debug logs for the SMTP server.

TCPWARE_SMTP_SERVER_RCPT_CHECK_HOST

The host name to be used in checking for local host when passing messages through the reject rules.

TCPWARE_SMTP_SERVER_REJECT_FILE

Points to the file containing the rejection rules.

TCPWARE_SMTP_SERVER_REJECT_INFO

Specifies the level of OPCOM messages generated by the rejection rules for incoming SMTP mail. If not defined, no messages are generated.

TCPWARE_SMTP_SUPPRESS_VENDOR

Suppresses the vendor name in the SMTP server welcome banner. Define this logical to hide the fact that the system is a VMS system running TCPware.

TCPWARE_SMTP_SYMBIONT_LOG

Enables debug logs for the SMTP symbiont.

TCPWARE_SMTP_SYMBIONT_PURGWS_TIMER

Specifies how often the SMTP symbiont purges its working set to free up unneeded memory. The time is specified as a delta time.

TCPWARE_SMTP_WINDOW_SIZE

Specifies the window size used in TCP connections when delivering mail.

TCPWARE_SNMP_DEBUG

SNMP subagent developers uses this logical to set certain debug masks.

```
$ DEFINE TCPWARE_SNMP_DEBUG mask
```

TCPWARE_SSH_ALLOW_EXPIRED_PW

Allows logging in to an account when the account's password has expired due to pwdlifetime elapsing. This applies to all users and circumvents normal VMS expired-password checking, and therefore should be used with caution. An entry is made into the SSH_LOG:SSHD.LOG file when access is allowed using this logical name.

TCPWARE_SSH_ALLOW_PREEXPIRED_PW

(SSH1) allows logging in to an account when the password has been pre-expired. This applies to all users and circumvents normal VMS expired-password checking, and therefore should be used with caution. An entry is made into the SSH_LOG:SSHD.LOG file when access is allowed using this logical name.

TCPWARE_SSH_KEYGEN_MIN_PW_LEN

(SSH1) defines the minimum passphrase length when one is to be set in SSHKEYGEN. If not defined, defaults to zero. Defined by @TCPWARE2CNFNET SSH.

TCPWARE_SSH_PARAMETERS_*n*

These parameters are used to start SSHD_MASTER. They are parameters set by @TCPWARE:CNFNET SSH.

TCPWARE_SSH_USE_SYSGEN_LGI

(SSH1) if defined, causes SSHD to use the VMS SYSGEN value of LGI_PWD_TMO to set the login grace time, overriding anything specified in the command line or the configuration file.

TCPWARE_SVCORDER

Contains the list of services used in the order specified.

Use the values `"bind,local"` (the default if the logical is not defined) and `"local,bind"` (uses DNS if the Hosts database lookup fails).

TCPWARE_VMSMAIL_HEADER_CONTROL

Specifies how many RFC822 headers are included in mail delivered to VMS Mail users. Values can be ALL, MAJOR, and NONE.

TCPWARE_VMSMAIL_LOCASE_USERNAME

Lowercases the username portion of outgoing addresses.

TCPWARE_VMSMAIL_NO_EXQUOTA

Delivers incoming mail to local VMS Mail users without using EXQUOTA.

TCPWARE_VMSMAIL_REPLY_CONTROL

Specifies which header to use to determine the sender of a message ("Reply-To:" or "From:").

TCPWARE_VMSMAIL_USE_RFC822_TO_HEADER

Sets the maximum length of the RFC822 To: header line when sending outgoing mail. The default is 1024. The range can be set from 256 to 65535.

TCPWARE_TCLB_BIAS

Define this logical with a multiplier and an addend as two values of the logical. Both are real numbers.

You can use these values to bias a load offered to the host. For example, the following command doubles the observed load and adds 1.5 users:

```
$ DEFINE /SYSTEM TCPWARE_TCLB_BIAS "2.0","1.5"
```

TCPware re-translates this logical before it sends each response. This means that some other process can change it dynamically or you can set it statically.

TCPWARE_TELNET_WINDOW

Specifies the window size that the TELNET server offers to the peer. The default value is 4096. If the value is less than 512, TELNET uses 4096.

TCPWARE_TELNETD_DEFCHAR

Sets up the default terminal characteristics for TELNET sessions. You can avoid having to change the SYSGEN TTY_DEFCHAR and TTY_DEFCHAR2 fields system-wide. This logical forces the hangup bit set. To prevent the forcing of the hangup bit set, use the TCPWARE_TELNETD_NO_FORCED_HANGUP logical.

TCPWARE_TELNETD_FLAGS

Setting either bit 0 or 1 can improve server performance and reduce system processing overhead. The default value is 1.

Note! Doing so means you are not adhering to the TELNET protocol.

TCPWARE_TELNETD_INTRO_MSG

Defines a special message that appears whenever a user attempts access to the host through TELNET. Use this logical to issue warnings such as "Authorized Use Only" for remote logins.

If the TCPware ACE/Client is enabled and the user is designated for Token Authentication, the user is also prompted for the PASSCODE in addition to the username and password.

Kerberos password protection is also available for the TELNET service.

TCPWARE_TIMED_EXCLUDE

Determines the networks excluded from clock synchronization, either in network addresses or names.

TCPWARE_TIMED_INCLUDE

Determines the networks included in clock synchronization, either in network addresses or names.

TCPWARE_TIMED_MODE

Determines if the current host is a MASTER, FIXED MASTER, or SLAVE.

- **MASTER** (primary) -- broadcasts time synchronization requests, calculates the time differences and averages, and sends "adjust time" messages.
- **FIXED MASTER** (fixed primary) -- provides absolute time stamps to newly started dependent TIMED hosts.
- **SLAVE** (dependent) -- is the recipient of primary "adjust time" messages.

TCPWARE_TIMEZONE

This logical can have two equivalence strings:

- *+hhmmss*

  hh are the hours *mm* are the minutes *ss* are the seconds offset from the universal time (UT).

  + is for east of the central meridian, – is for west. For example: `+04:00:00` is four hours east of the central meridian at Greenwich.

  Another example: eastern standard time (EST) is five hours west of UT, so the offset is `–0500`.

- *name* an optional name for the time zone. For example: EDT for Eastern Daylight time. Can be one of the following:

  Universal Time--UT, UTC, or GMT
  North American Time--EST, EDT, CST, CDT, MST, MDT, PST, PDT
  Military Time--Any single uppercase letter A through Z except J (this format is not recommended)

  Any other character sequence

  The *name* is not validated and may be used by applications to report the local time zone.

TCPWARE_TSSYM_*qname*

Defines the parameters normally set with the /ON qualifier. Since you cannot use /AUTOSTART_ON together with the /ON qualifier to initialize a terminal server print queue, you need to define TCPWARE_TSSYM_*qname* for this purpose.

```
$ DEFINE /SYSTEM TCPWARE_TSSYM_qname "host,port[,option...]"
```

TCPWARE_TSSYM_*_ RETRY_INTERVAL

Defines the interval at which the symbiont retries to make a connection to a printer after an attempt fails. The default is 0::15 (15 seconds delta time).

TCPWARE_TSSYM_*_TIMEOUT

Defines the time it takes for a print job to abort if the connection to the printer is never established. The default timeout is infinite (it never times out).

TCPWARE_TSSYM_*qname*_RETRY_INTERVAL

Same as TCPWARE_TSSYM_*_RETRY_INTERVAL, but for a specific queue only, and overrides TCPWARE_TSSYM_*_RETRY_INTERVAL.

TCPWARE_TSSYM_*qname*_TIMEOUT

Same as TCPWARE_TSSYM_*_TIMEOUT, but for a specific queue only, and overrides TCPWARE_TSSYM_*_TIMEOUT.

TCPWARE_VMSLPRSMB_ *qname*_PRECONN

Makes the connection to the printer *before* processing the file. Normal behavior is to make the connection to the printer *after* processing the file.

TCPWARE_VMSLPRSMB_*qname*_RETRY_INTERVAL

Same as TCPWARE_VMSLPRSMB_*_RETRY_INTERVAL, but for a specific queue only, and overrides TCPWARE_VMSLPRSMB_*_RETRY_INTERVAL.

TCPWARE_VMSLPRSMB_*qname*_TIMEOUT

Same as TCPWARE_VMSLPRSMB_*_TIMEOUT, but for a specific queue only, and overrides TCPWARE_VMSLPRSMB_*_TIMEOUT.

TCPWARE_VMSLPRSMB_*_RETRY_INTERVAL

Defines the interval at which the symbiont retries to make a connection to a printer after an attempt fails. The default value for a retry interval is 2 minutes (:2 in delta time).

Note! A connection failure can take 1.5 minutes to time out, which is not included in this interval value.

TCPWARE_VMSLPRSMB_*_TIMEOUT

Defines the time it takes for a print job to abort if the connection to the printer is never established. The default timeout is infinite (it never times out).

UCX$DEVICE

Defined as `BG:` (the name of the UCX device drive).

UCX$INET_HOST

Defined to be the host name (the same setting as TCPWARE_DOMAINNAME).

UCX$IPC_SHR

Provides the linkage to the TCPware version of the UCX$IPC_SHR Run-Time library.

Appendix C SSH Status Codes

This appendix has tables that show the status codes for the following SSH clients: SSH2, SSH-ADD2, SSH-KEYGEN, SSH-CMPCLIENT, SSH-CERTTOOL, SSH-CERTVIEW, SCP2, and SFTP2.

SSH Client Status Codes

The following table shows the new status codes for the following SSH clients: SSH2, SSH-ADD2, SSH-KEYGEN, SSH-CMPCLIENT, SSH-CERTTOOL and SSH-CERTVIEW clients. These codes are implemented in TCPware V5.x.

To enable these status code instead of using the pre-TCPware V5.x codes, the logical name TCPWARE_SSH_NEW_STATUS_CODES must be defined system-wide

Table C-1 SSH Status Codes Sorted by Name

Error Code	Error Name	Description
0C1F8044	AGENTBADPASS	Invalid password entered
0C1F804C	AGENTERROR	General error
0C1F806A	AGENTNOAGENT	No agent is available
0C1F8072	AGENTNOFILE	Private key is unreadable
0C1F807A	AGENTNOID	Key not found in authentication agent
0C1F83F1	AGENTOK	Successful operation by agent
0C1F8082	AUTHCANCEL	Authentication cancelled by user
0C1F803C	AUTHFAIL	Authentication failed
0C1F808A	CERT12ENCOD	Certificate PKCS#12 encoding failed
0C1F8092	CERT12SAVE	Failed to save PKCS#12 package

0C1F809A	CERTBADSTATUS	Bad status returned
0C1F80A2	CERTCANTSETPUB	Failed to set public key
0C1F80AA	CERTERROR	Certificate error
0C1F80B2	CERTNO10SIGN	No PKCS#10 requests signed
0C1F80C2	CERTNOSER	No serial number supplied
0C1F80CA	CERTNOVAL12OBJ	No objects to store in PKCS#12 package
0C1F80D2	CERTPRVKEYGEN	Failed to generate private key
0C1F80DA	CERTPRVKEYREAD	Failed to read private key
0C1F80E2	CERTPRVKEYWRT	Failed to write private key
0C1F80EA	CERTUNDEF	Undefined error
0C1F80F2	CERTWRTFILEB64	failed to write base64 file
0C1F80FA	COMPERR	Compression error
0C1F8102	CONNECTFAIL	Connection failed
0C1F80BA	CONNNOTALLOWED	Connection not allowed
0C1F810A	DISCONBYAPP	Session disconnected by application
0C1F8112	E2BIG	Argument list too long
0C1F811A	EABANDONED	Owner can not release resource
0C1F8122	EACCES	Permission denied
0C1F812A	EADDRINUSE	Address already in use
0C1F8132	EADDRNOTAVAIL	Can't assign requested address
0C1F813A	EAFNOSUPPORT	Address family not supported
0C1F8142	EAGAIN	No more processes

0C1F814A	EALIGN	Alignment error
0C1F8152	EALREADY	Operation already in progress
0C1F815A	EBADCAT	Bad message catalogue format [1]
0C1F8162	EBADF	Bad file number
0C1F816A	EBADMSG	Corrupted message detected
0C1F8172	EBUSY	Mount device busy
0C1F817A	ECANCELED	Operation canceled
0C1F8182	ECHILD	No children
0C1F818A	ECONNABORTED	Software caused connection abort
0C1F8192	ECONNREFUSED	Connection refused
0C1F819A	ECONNRESET	Connection reset by peer
0C1F81A2	EDEADLK	Resource deadlock avoided
0C1F81AA	EDESTADDRREQ	Destination address required
0C1F81B2	EDOM	Math argument
0C1F81BA	EDQUOT	Disk quota exceeded
0C1F81C2	EEXIST	File exists
0C1F81CA	EFAIL	Cannot start operation
0C1F81D2	EFAULT	Bad address
0C1F81DA	EFBIG	File too large
0C1F81E2	EFTYPE	Inappropriate operation for file type
0C1F81EA	EHOSTDOWN	Host is down
0C1F81F2	EHOSTUNREACH	No route to host

0C1F81FA	EIDRM	Identifier removed
0C1F8202	EILSEQ	Illegal byte sequence
0C1F820A	EINPROG	Asynchronous operation in progress
0C1F8212	EINPROGRESS	Operation now in progress
0C1F821A	EINTR	Interrupted system call
0C1F8222	EINVAL	Invalid argument
0C1F822A	EIO	I/O processing error
0C1F8232	EISCONN	Socket is already connected
0C1F823A	EISDIR	Is a directory
0C1F8242	ELOOP	Too many levels of symbolic links
0C1F824A	EMFILE	Too many open files
0C1F8252	EMLINK	Too many links
0C1F825A	EMSGSIZE	Message too long
0C1F8262	ENAMETOOLONG	File name too long
0C1F826A	ENETDOWN	Network is down
0C1F8272	ENETRESET	Network dropped connection on reset
0C1F827A	ENETUNREACH	Network is unreachable
0C1F8282	ENFILE	File table overflow
0C1F828A	ENOBUFS	No buffer space available
0C1F8292	ENODEV	No such device
0C1F829A	ENOENT	No such file or directory
0C1F82A2	ENOEXEC	Exec format error

0C1F82AA	ENOLCK	No locks available
0C1F82B2	ENOMEM	Not enough core
0C1F82BA	ENOMSG	No message of desired type
0C1F82C2	ENOPROTOOPT	Protocol not available
0C1F82CA	ENOSPC	No space left on device
0C1F82D2	ENOSYS	Function not implemented
0C1F82DA	ENOTBLK	Block device required
0C1F82E2	ENOTCONN	Socket is not connected
0C1F82EA	ENOTDIR	Not a directory
0C1F82F2	ENOTEMPTY	Directory not empty
0C1F82FA	ENOTSOCK	Socket operation on non-socket
0C1F8302	ENOTSUP	Function not implemented
0C1F830A	ENOTTY	Not a typewriter
0C1F8312	ENWAIT	No waiting processes
0C1F831A	ENXIO	No such device or address
0C1F8322	EOPNOTSUPP	Operation not supported on socket
0C1F832A	EPERM	Not owner
0C1F8332	EPFNOSUPPORT	Protocol family not supported
0C1F833A	EPIPE	Broken pipe
0C1F8342	EPROCLIM	Too many processes
0C1F834A	EPROTONOSUPPORT	Protocol not supported
0C1F8352	EPROTOTYPE	Protocol wrong type for socket

0C1F835A	ERANGE	Result too large
0C1F8362	EREMOTE	Too many levels of remote in path
0C1F836A	EROFS	Read-only file system
0C1F8372	ESHUTDOWN	Can't send after socket shutdown
0C1F837A	ESOCKTNOSUPPORT	Socket type not supported
0C1F8382	ESPIPE	Illegal seek
0C1F838A	ESRCH	No such process
0C1F8392	ESTALE	Stale NFS file handle
0C1F839A	ETIMEDOUT	Connection timed out
0C1F83A2	ETOOMANYREFS	Too many references: can't splice
0C1F83AA	ETXTBSY	Text file busy
0C1F83B2	EUSERS	Too many users
0C1F83BA	EWOULDBLOCK	Operation would block processing to complete
0C1F83C2	EXDEV	Cross-device link
0C1F8014	EXECERR	Subprocess execution error
0C1F800C	FATALERR	Fatal error
0C1F805A	HOSTNOTALLOW	Host not allowed to connect
0C1F8024	ILLUSER	Illegal username
0C1F801C	KEYEXFAILED	Key exchange failed
0C1F802C	KEYNOTVER	Key not verified
0C1F8034	MACERR	MAC error
0C1F8062	NOMOREMETH	No more authentication methods

0C1F83CA	PROTERR	Protocol error
0C1F83D2	PROTNOTSUP	Protocol not supported
0C1F83DA	SRVNOTAVAIL	Service not available
0C1F83E9	SUCCESS	Successful completion
0C1F8052	TOOMANYCONN	Too many connections
0C1F83E2	UNDEFDISCONCODE	Undefined disconnect reason

SFTP2 Client Status Codes

The following table shows the status codes for the SFTP2 file transfer client.

Table C-2 SFTP2 Status Codes Sorted by Name

Error Code	Error Name	Description
0C1F8092	BAD_BUFSIZE	BUFFER_SIZE cannot be less than 512
0C1F809A	BAD_CONCUR	Concurrent_requests requires an argument greater than zero
0C1F807A	BAD_DEBUG	Debug value is out of range
0C1F804A	BAD_DEST	Invalid destination specification
0C1F802A	BAD_PORT_NUM	Port specification is bad or out of range
0C1F8022	BAD_QUALIFIER	Unrecognized command qualifier
0C1F803A	BAD_SOURCE	Invalid source specification
0C1F8082	BAD_TRANSLATE	Bad combination of values for /TRANSLATE_VMS
0C1F800C	CHILD_DIED	SSH2 child process died unexpectedly
0C1F8062	CONNECTION_ERR	Unable to establish or maintain connection to remote system

0C1F805A	DEST_NOT_DIR	Destination is not a directory
0C1F8018	FILE_OVERWRITTEN	Existing file overwritten
0C1F8014	INTERNAL_ERROR	SFTP2 Fatal internal error
0C1F8032	MISSING_DEST	Destination file specification is missing
0C1F8072	NO_PERMISSION	Permission denied
0C1F806A	NO_SUCH_FILE	No such file
0C1F8052	PROTO_ERR	Protocol errors
0C1F8042	SOURCE_NOT_AVAIL	Unable to open source file
0C1F80A1	SUCCESS	Successful completion
0C1F808A	TRANSFER_ERR	Error transferring file

SCP2 Client Error Codes

The following table shows the status codes for the SCP2 file transfer client:

Table C-3 SCP2 Status Codes Sorted by Name

Error Code	Error Name	Description
0C1F809A	BAD_BUFSIZE	BUFFER_SIZE cannot be less than 512
0C1F80A2	BAD_CONCUR	Concurrent_requests requires an argument greater than zero
0C1F8082	BAD_DEBUG	Debug value is out of range
0C1F8052	BAD_DEST	Invalid destination specification
0C1F80AA	BAD_OFFSET	Bad offset for READOFFSET or WRITEOFFSET
0C1F8032	BAD_PORT_NUM	Port specification is bad or out of range

0C1F802A	BAD_QUALIFIER	Unrecognized command qualifier
0C1F8042	BAD_SOURCE	Invalid source specification
0C1F808A	BAD_TRANSLATE	Bad combination of values for /TRANSLATE_VMS
0C1F800C	CHILD_DIED	SSH2 child process died unexpectedly
0C1F806A	CONNECTION_ERR	Unable to establish or maintain connection to remote system
0C1F8062	DEST_NOT_DIR	Destination is not a directory
0C1F8018	FILE_OVERWRITTEN	Existing file overwritten
0C1F8014	INTERNAL_ERROR	SCP2 Fatal internal error
0C1F803A	MISSING_DEST	Destination file specification is missing
0C1F807A	NO_PERM	Permission denied
0C1F8020	NO_PERMISSION	Permission denied
0C1F8072	NO_SUCH_FILE	No such file
0C1F805A	PROTO_ERR	Protocol errors
0C1F804A	SOURCE_NOT_AVAIL	Unable to open source file
0C1F80B1	SUCCESS	Successful completion
0C1F8092	TRANSFER_ERR	Error transferring file

Glossary

This appendix provides a glossary of terms found throughout the TCPware for OpenVMS documentation set.

Glossary of Terms

access control list (ACL)	OpenVMS list containing access rights for users.
access restrictions	Restrictions on a TCP application's usage, either incoming or outgoing.
active open	Actively opens a connection. TCPDRIVER sends segments to establish a connection to the destination host and port number for an active open request. To establish a connection, a passive open must usually be pending on the destination host.
Address Resolution Protocol (ARP)	Protocol used to map internet addresses to physical hardware addresses used on Ethernet and FDDI. *See* Fiber Distributed Data Interface and Reverse Address Resolution Protocol.
Ancillary control process (ACP)	A process that acts as an interface between user software and an I/O driver. An ACP provides functions supplemental to those performed in the driver, such as file and directory management.
application program interface (API)	Programming interface to an application, such as the TCPware SNMP Extendible Agent MIB API, ACE/Client API for Token Authentication, or the interface between a terminal emulation program and the TES Client software.
ARPANET	First entity to implement TCP/IP. ARPANET is the DARPA internet that served as the backbone for TCP/IP research. TCP/IP was so successful in the ARPANET that DARPA designated TCP/IP as a networking standard.
Asynchronous Transfer Mode (ATM)	*See* Classical IP over ATM.
attributes data file (ADF)	Special file in the NFS Client that maintains the attributes for an OpenVMS data file. These files appear on the server as `.$ADF$`*`filename`*, although the client system cannot see them.

authenticator	The Kerberos protocol uses authenticators to prevent eavesdroppers from stealing a ticket. The client sends a new authenticator with each request for service from a server. An authenticator consists of the client's name, client's IP address, and a timestamp showing the current time.
automounting	Automatic and transparent mount in NFS that mounts a filesystem when accessing it.
Autonomous System (AS)	Set of routers under a single technical administration, using an internal protocol and common metrics to route packets within the AS, and an external protocol to route packets to other ASs. The NIC assigns AS numbers.
background mount	Attempts to mount a filesystem on the NFS client made at least once at varying intervals and specified number of retries. *See* multicasting.
big-endian	Format for storage of binary data where the most-significant byte comes first. The Internet's standard byte order is big-endian. *See* little-endian *and* network byte order.
Border Gateway Protocol (BGP)	Exterior routing protocol used to exchange routing information between multiple transit Autonomous Systems (ASs) as well as between transit and stub ASs.
broadcasting	Packet delivery system that provides a copy of a given packet to all hosts attached to the network. For example: Ethernet and FDDI. *See* multicasting.
Classical IP over Asynchronous Transfer Mode (CLIP)	A way of sending IP datagrams over ATM protocol lines.
Classless Inter Domain Routing (CIDR)	Protocol developed in 1992 by the Internet Engineering Steering Group that eliminates address class distinctions and depends on address masks that fall on bit instead of byte boundaries. The strategy assigns blocks of Class C addresses to Internet providers and has the providers "subnet mask" the addresses in further units to organizations. This also sharply reduces the growth in routing tables in Internet routers beyond their manageable capacity.
client-server model	Concept used to describe the application layer protocols. The process that initiates a service is the client (or user). The process that provides the service is the server. A client and a server can be on different hosts or on the same host.

cluster alias failover	System whereby a node in a VMScluster (the "alias") can accept incoming connection requests for a server if the servicing node goes down. Used primarily with the Network File System (NFS).
Compressed SLIP (CSLIP)	*See* Serial Line IP (SLIP).
connectionless service	Service that presents data complete with a destination address, and the network delivers it on a best-effort basis, independent of other data exchanged between the same pair of users. Examples include IP and UDP.
connection-oriented service	Service that implements a connection-setup procedure before it can exchange data between two users. Connection-oriented services or protocols provide data transfer that is reliable, ordered, full-duplex, and flow-controlled. TCP is a connection-oriented service.
data circuit-terminating equipment (DCE)	Term the X.25 protocol standards use that applies to switching equipment that forms a packet switched network to distinguish it from the computers or terminals that connect to the network.
datagram	Single message unit IP uses over an internet and consisting of protocol headers and data.
data terminal equipment (DTE)	Term X.25 protocol standards use that applies to computers and/or terminals, to distinguish them from the packet switching network to which they connect.
delta time	The delta time syntax is: dddd hh:mm:ss.cc • *dddd* is the number of days (0-9999); if less than one day, specify zero (0); follow with a blank space. • *hh* is the number of hours (0-23). • *mm* is the number of minutes (0-59) preceded by a colon (:). • *ss* is the number of seconds (0-59) preceded by a colon (:). • *cc* is the number of hundredths of a second (0-99) preceded by a period (.). You can truncate a delta time on the right. You can omit fields in the time format as long as you include the punctuation that separates the fields. You must specify the days field even if you omit all time fields.

domain namespace	Naming hierarchy. A domain name consists of a sequence of names (labels) separated by periods (.). The following are examples of domain names: NS.NASA.GOV C.NYSER.NET BBN.COM
Domain Name System (DNS)	System that allows access to a distributed, hierarchical database of internet addresses, hostnames, and other information throughout the Internet.
duress PIN	Special PIN to use if you are being compromised during the login process. Used with the token authentication system.
Dynamic Host Configuration Protocol (DHCP)	Protocol that centralizes and automates TCP/IP network configuration. The DHCP Server dynamically allocates IP addresses for hosts on the network from an available pool of addresses. In this way, new hosts or hosts that are frequently relocated can automatically get new IP addresses for a certain lease period. DHCP is an extension of the Internet Bootstrap Protocol (BOOTP).
dynamic routing	*See* Gateway Routing Daemon.
encryption	Transformation of plain text into unintelligible text.
EXPORT database	Database on the NFS server system that controls which filesystems the server is able to export to a client.
exporting	Making a network filesystem available to mount on a client system by listing it in the "export" database.
Exterior Gateway Protocol (EGP)	Exterior routing protocol that moves routing information between Autonomous Systems (ASs).
External Data Representation (XDR) Protocol	Standard that resolves differences of data representation between different operating systems and hardware architectures.
Fiber Distributed Data Interface (FDDI)	Set of ANSI/ISO standards that define a high-bandwidth (100 Mb/s) general-purpose LAN. It provides synchronous and asynchronous services between computers and peripheral equipment in a timed-token passing, dual ring of trees configuration.
File Sharing Services (FSS)	NetWare service that lets you access OpenVMS directories, files, and printers using DOS facilities.

File Transfer Protocol (FTP)	Application level protocol that allows a user on a client host to log in to a server host and perform file functions.
filename mapping	Process in NFS of mapping filenames between OpenVMS and UNIX so as to preserve the respective systems' file naming conventions.
filesystem	Method for recording, cataloging, and accessing files on a client or server system.
flat namespace	In flat namespace naming, a system selects object names from a single set of strings rather than a hierarchical organization of strings. The following hostnames are examples: ALPHA RESEARCH TULIP
Gateway Routing Daemon (GateD)	Manages multiple routing protocols, including the Routing Information Protocol (RIP), Local Network Protocol (HELLO), Router Discovery Protocol, Open Shortest Path First (OSPF) protocol, Exterior Gateway Protocol (EGP), and Border Gateway Protocol (BGP).
gateway	Device used to connect two or more networks to form an internet. A gateway also has an internet address for each connected network, and performs routing functions.
GROUP database	Database on the NFS Client that authorizes a client's group access to the remote host's filesystems. The database contains the group number and the VMS group identifier corresponding to the remote group identifier in the UNIX `/etc/group` file.
group ID (GID)	Group identification on the UNIX NFS host.
HELLO	Also called the Local Network Protocol, it is an interior protocol that uses delay as the deciding factor when selecting the best route. Delay is the round-trip time between source and destination. HELLO is not currently widely in use.
host	Unique, addressable entity that is part of an internet. A multiuser minicomputer and a terminal server are examples of hosts.
host byte order	Standard a host uses for storage and transmission of integers that specifies that either the least significant byte or most significant byte appears first. Sending machines must translate from their host or local machine integer representation to network byte order. Receiving machines must translate from network byte order to the local host or local machine representation. *See* big-endian *and* little-endian.

host equivalence files	Security access files on a Berkeley R Commands server host used to authorize access to services by other hosts or users. The files list hostnames (and, optionally, usernames) and indicate which remote hosts and users have equivalent access as local users. These include `RHOSTS.` and `HOSTS.EQUIV` files.
hostname	Name assigned to a host. These names are for user convenience and a system maps it to an internet address. Host names may either be from a flat namespace or the domain namespace. A hostname is a mnemonic given to a host for the purpose of identifying it. Because the TCP/IP protocols only understand internet address, they must "translate" these hostnames into internet addresses. TCPware supports two means for translating a hostname into an internet address (or vice versa); the `HOSTS.` file and Domain Name System (DNS). The `HOSTS.` file supports any naming conventions you wish to use. Typically, use of a HOSTS. file involves using a flat namespace. For larger networks and the Internet, systems now more commonly use the Domain Name System (DNS).
idempotency	Remote Procedure Call (RPC) jargon for performing an operation more than once with identical results and without causing any harm. For example, an NFS server receives a delete file request from a client, deletes the file, and sends a success reply, but the network loses the reply before it reaches the client. Because the client does not receive a reply, it sends the delete file request again. Rather then process the request again and send a false error message stating that the file does not exist, the server simply retransmits the original reply.
instance	In Kerberos authentication, identifies an instantiation of a principal name, such as the name of the system running a server.
internet	Network formed by connecting dissimilar hosts and networks with TCP/IP protocols. When capitalized (Internet), this term refers to the ARPANET, the DARPA internet that forms the backbone of internet research.
internet address	Unique 32-bit value assigned to each host in an internet. All internet communications with a particular host use its internet address. TCPDRIVER, UDPDRIVER, IPDRIVER, INETDRIVER, and BGDRIVER use internet addresses to identify a host on the network. Each host on the network assumes a unique internet address. Internet addresses are 32-bit values. Internet addresses are in reverse VAX byte order (the most significant byte of the internet address is in the least significant byte of the longword value).

Internet Control Message Protocol (ICMP)	Performs a function of IP by providing a communications facility for gateways and hosts.
Internet Message Access Protocol (IMAP)	Allows IMAP-compliant mail programs to access messages stored remotely as if the storage were local.
Internet Protocol (IP)	Basis of TCP/IP, providing the network interface and message routing services for the higher level protocols.
Internet Printing Protocol (IPP)	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.
IP routing	Mechanism provided by IP to deliver datagrams from the source to the destination.
IP Security Option (IPSO)	U.S. Department of Defense standard for protecting datagrams over the network.
Kerberos	Authentication system for open systems and networks. Kerberos uses a set of encrypted keys and tickets for authentication. Kerberos provides network security by regulating user access to networking services.
Key Distribution Center (KDC)	An alternate name for the Kerberos Server.
layer	The TCP/IP protocol suite consists of three layers of services that rest on a layer of hardware.
little-endian	Format for storage of binary data in which the least significant byte comes first. The VAX, Alpha and I64 byte order is little-endian. *See also* big-endian.
load balancing	Also known as TCP/IP load balancing. The system whereby the server changes the preferred order of access to systems in a TCP/IP cluster in response to their observed load.
local area network (LAN)	Two or more hosts connected by the same communications medium. The hosts typically span a small geographic area such as a single room or building.
Management Information Base (MIB-II)	Most recent MIB version for the SNMP protocol. A collection of data residing on the SNMP agent host and organized into groups. Each piece of data within a group is a management object.

mask, address or network	32-bit internet address, where the network number is set to all bits one and the host number is set to all bits zero. Hosts and gateways use the network mask to route internet packets by extracting the network number of an internet address, and comparing the network number with their own routing information to determine if the packet is bound for a local address.
master file directory (MFD)	"Root" directory ([000000]) in OpenVMS that is the default mount point for an NFS filesystem.
mount	NFS protocol that provides file handles for server access and keeps track of mounts.
mount point	Point on the remote NFS directory tree that you are interested in mounting or the point on the local directory tree where the remote filesystem is "attached."
mounting	Process in NFS of "attaching" a server filesystem to the file structure of a client to make it accessible using the client's normal operating facilities.
multicasting	Special form of broadcasting that delivers copies of the packet to only a subset of all possible destinations. *See* broadcasting.
multiplexing	Transmission of a number of different messages simultaneously over a single circuit.
multithreading	Ability to service transactions from many clients simultaneously.
network	Element of an internet in which two or more hosts are connected with the same communications medium. A LAN is an example of an internet network element.
network byte order	Internet standard for transmission of integers that specifies most significant byte appears first. Sending machines must translate from the local integer representation to network byte order, and receiving machines must translate from network byte order to the local machine representation. Equivalent to big-endian. *See* little-endian.
Network Control Utility (NETCU)	TCPware's utility program system managers and operators use to configure and control networks that run TCPware.
Network File System (NFS)	Application layer protocol developed by Sun Microsystems, Inc. that provides access to a remote computer's files as if they were local files.

Network Information Center (NIC)	Central organization of a network with the authority to create network names and addresses. NIC.DDN.MIL is the specific Internet NIC that holds the authority to create root servers.
Network Lock Manager (NLM) and Status Monitor (NSM)	The way in which the Network File System (NFS) supports file locking. Many NFS client systems support file locking, even on the record and byte level, as long as the byte ranges do not overlap. File locking on the Server is multithreaded, where the Server can satisfy more than one lock request at a time. The NSM cooperates with other status monitors on the network to notify the NLM of any changes in system status (such as when a crash occurs).
Network Print Services (NPS)	NetWare service that lets OpenVMS users print their files on any printer connected to NetWare LANs.
Network Time Protocol (NTP)	Protocol that synchronizes timekeeping among a set of distributed time servers and clients.
NSLOOKUP	Utility that queries information from DNS servers, based on RFCs 1034 and 1035.
occluded mount	Action in NFS where a filesystem mounts on a subdirectory of an existing mount point so that previously visible subdirectories and files of the original mount are no longer visible.
ONC RPC Services	Software development tool with which programmers can build distributed applications on VAX computers.
Open Shortest Path First (OSPF) Protocol	Interior gateway protocol that distributes routing information between routers in a single Autonomous System (AS). OSPF chooses the least cost path as the best path.
overmounting	Action in NFS where a filesystem mounts on top of an existing mount point.
packet	Single message as it appears to the physical network.
packet filtering	Restricts the datagrams that an interface can receive.
Packet Switching Data Network (PSDN) and packet switching exchange (PSE)	A PSDN consists of widely separated packet switching exchanges (PSEs). PSEs connect through public or private telephone networks or leased lines. PSEs contain data circuit-terminating equipment (DCE).
Passcode	Combination of your PIN and the tokencode. Used with the token authentication system.

passive open	Passive open "listens" and waits for a request from a remote host to establish a connection. You can fully or partially specify passive opens. Use partially specified opens when you do not know the requesting host. Note that a passive open does not send requests to establish a connection until the system receives a request from another host.
pathname	Directory path in a remote NFS filesystem.
PCNFSD	NFS authentication server to allow remote printing over NFS.
PC-peer	Synchronized host in the Network Time Protocol (NTP), which is either a time server or client and is identified by a relative NTP strata number.
PIN	Your personal identification number and part of the token authentication system. The PIN consists of four to eight alphanumeric characters.
PING	Packet InterNet Groper, a utility that tells you whether a host is up and whether you can reach it. The PING utility uses the ICMP echo and echo reply messages.
Point-to-Point Protocol (PPP)	Protocol whereby you can send IP datagrams over serial links, including LAT or modem connections. PPP is an enhancement to the nonstandard Serial Line IP (SLIP) interface, providing self-contained error detection and automatically negotiated header compression. It also provides authentication through the Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP).
port and port number	Abstract point through which a datagram passes from the host layer to the application layer protocols. Port number is a number the network drivers use to name the ends of logical connections. Port numbers are 16-bit values. Some standard server port numbers are 21 for FTP, 23 for TELNET, and 25 for SMTP. Servers generally use the port numbers from 0 to 255. User port numbers start at 1024. Specify port numbers in normal VAX byte order, unless indicated otherwise.
Post Office Protocol Version 3 (POP3)	Multithreaded server that can handle up to 31 simultaneous client connections. POP3 does not perform any mail delivery functions but simply allows clients (mostly PCs) to retrieve new mail from local inboxes.

principal	Kerberos client and server names in the format `name.instance@realm`. For clients, *name* is the user's login name; for servers, name is the service name. *See* instance and realm.
print symbiont	Privileged process used to manage a queue of jobs sent to a local or remote printer.
Product Authorization Key (PAK)	HP's product licensing mechanism.
protocol	Standard that defines how computers on a network communicate with each other.
PROXY database	Database on the NFS client or server system that authorizes a client's access to the remote host's filesystems. The database contains the UNIX identity of its client, consisting of a UID and GID.
Quote-of-the-Day service (QUOTED)	TCP-based character generator service that listens for TCP connections on TCP port 17. Once you establish a connection, the service sends a short message. The service then throws away any data it receives and closes the connection.
realm	In Kerberos authentication, the name of a group of machines, such as those on a LAN, identifying the Kerberos administrative domain.
Record Management Services (RMS)	Set of operating system procedures called by programs to process files and records within files. Defines rules about how to store records in files.
Remote Compact Disk (RCD)	Utility that provides access to CD-ROM drives on remote TCP/IP systems.
Remote Copy Program (RCP)	UNIX-like command with which you can copy files over the network.
Remote Login Protocol	*See* TELNET.
Remote Magnetic Tape (RMT)	Utility that provides access to magnetic tape drives on remote TCP/IP systems.
Remote Procedure Call (RPC)	Set of protocols developed by Sun Microsystems, Inc. These protocols allow programs to invoke procedures on remote hosts as if the procedures were local. See ONC RPC Services.

Request for Comments (RFC)	Documents submitted to the Internet governing board to define Internet standards.
resolver	A Domain Name System (DNS) client that communicates with a DNS server to resolve a host name and internet address. The client does not maintain a database. The client only sends queries; it does not answer them.
resource record	Entry in a Domain Name System (DNS) database files.
Reverse Address Resolution Protocol (RARP)	Protocol used to map the physical hardware addresses to the IP address (used on Ethernet and FDDI). Diskless machines use this protocol to find their IP addresses from the server.
rlogin	Remote login; a Berkeley UNIX system service that allows users of one machine to connect to other UNIX machines across the Internet and interacts as if their terminals were directly connected to the machines. The software passes information about the user's environment, such as terminal type, to the remote machine.
Router Discovery Protocol	IETF standard protocol used to inform hosts of the existence of routers without having hosts wiretap routing protocols such as RIP. Used in place of, or in addition to statically configured default routes in hosts.
Routing Information Protocol (RIP)	Distance-vector protocol for distributing routing information at the local network level of the Internet. In distance-vector routing, each router transmits destination addresses and costs to its neighbors.
Serial Line IP (SLIP)	A point-to-point protocol used when you need to route TCP/IP traffic over a serial line instead of an Ethernet cable. You most commonly use SLIP to connect systems on two Ethernet networks some distance apart. Compressed SLIP (CSLIP) is used to compress the TCP/IP headers only (and not the data) over the SLIP line.
server	Host providing a service in a relationship between two cooperating processes.
Simple Mail Transfer Protocol (SMTP)	Application layer protocol that provides an electronic mail facility to an internet.
Simple Network Management Protocol (SNMP)	Allows network management stations to obtain timely information about the network activities of OpenVMS server hosts. The information describes such things as routing, line status, the volume of network traffic, and error conditions.

sliding window	Characteristic of protocols that allow the sender to transmit up to n packets before an acknowledgment arrives. After the system receives an acknowledgment for the first packet, the sending protocol slides the packet window along the stream and sends another packet.
socket	Abstraction first provided by Berkeley BSD UNIX that allows a process to have access to the Internet. A process opens a socket, specifies the desired service (reliable stream delivery, datagram delivery, IP) connects the socket to a specified destination, and then sends or receives data.
Socket Library	Collection of VAX C (on VAX machines) and DEC C (on Alpha and I64 machines) subroutines that closely emulates the UNIX socket functions.
SSH	Abbreviation for Secure Shell. *See* Accessing Remote Systems with the Secure Shell (SSH) Utilities.
statelessness	Ability not to have to maintain information from a previous request to process a new one. The Network File System (NFS) is an example of a stateless operation.
stratum	Number for a peer in the Network Time Protocol (NTP) that identifies the relative hierarchy of the peer. Lower strata peers act as time servers while higher strata peers are clients who adjust their time clocks according to these servers. An Internet Time Server (ITS) on the network is assigned *stratum 1* because it has radio-clock-generated time based on Universal Coordinated Time (UTC).
stream service	TCP service that transfers data in a continuous flow, without the use of markers to show the beginning or end of messages.
STREAM-LF file	Record structure OpenVMS uses where it views the file's records as a continuous stream of bytes delimited by a line feed (LF) character.
subnetwork (subnet)	Subdivision of a network used to provide a means to logically group hosts within a large network.
subnet mask	Thirty-two-bit internet address created by taking bits from the host number and using them to extend the network mask. Hosts and gateways local to a subnet use the subnet mask for local routing.
superuser	UNIX or NFS user having almost unlimited privileges. The superuser usually has a User ID (UID) of 0.

symbiont	Process that transfers record-oriented data to or from a device. For example, an output symbiont transfers data from disks to line printers. See print symbiont.
TALK	Utility that allows users to exchange messages they type in their terminal windows with other local or remote users.
TELNET	Application layer protocol that allows a user at a client host to log in to a server host. The user's terminal at the client host appears to the server as a directly connected terminal.
Terminal Emulation Services	Transport protocol that provides NetWare workstations access to any OpenVMS systems.
Terminal Server Print Services	Provides an efficient way for OpenVMS users to send print requests to printers attached to TCP/IP-based terminal servers. Users on the host can easily gain access to printers attached to a terminal server as if they were any other OpenVMS printer.
ticket	Kerberos authentication entity that allows a user to prove his identity to an application server by way of a third-party (Kerberos) server.
Time Synchronization Protocol (TSP), or TIMED	Protocol that synchronizes the clocks of the various hosts in a LAN. Also know as `timed`.
TN3270	Mode used in TELNET to communicate with IBM 3278-*n* terminal models.
token authentication	An authentication system that allows you to set additional security restrictions on your FTP, TELNET, RLOGIN, and SET HOST logins. Authentication takes place through a physical SecurID token "smart card" that you use to provide the token authentication server (ACE/Server) with the necessary login information. You can set up token authentication through TCPware's Access Control Encryption Client (ACE/Client) on the OpenVMS host, which communicates with Security Dynamics' ACE/Server on a UNIX or Windows NT host.
tokencode	Random number currently displayed on your Security Dynamics SecurID smart card. Used with the token authentication system.
transaction service	Method of data transport provided by UDP that treats each datagram as a separate entity.

Transmission Control Protocol (TCP)	Host layer protocol that provides a reliable data transport service to the application layer protocols. TCP is stream-oriented. It ensures that the system delivers data in order and without duplication.
transparency	Level at which a user need not be aware of the process involved but only in the results of an operation.
trap	Unsolicited message the SNMP agent sends to a management station to inform it that a change in the network occurred. *See also* Simple Network Management Protocol (SNMP).
UNIX or ULTRIX filesystem	Set of files organized as a tree with a single root node (`root`) indicated as a slash (/).
User Datagram Protocol (UDP)	Host layer protocol that provides transaction oriented data transport. UDP does not provide data reliability, but does provide data transport with very little overhead.
user ID (UID)	User identification on the UNIX Network File System (NFS) host.
User Identification Code (UIC)	User identification on the OpenVMS host in the format *`username`* or [*`group,member`*].
VAX byte order	VAX standard for storage and transmission of integers that specifies that the least significant byte appears first. VAX byte order is little-endian. VAX byte order sending machines must translate from the local integer representation to network byte order, and receiving machines must translate from network byte order to the local machine representation.
virtual circuit	Facility in a packet-switched communication network in which packets passing between a pair of terminals stay in sequence. Since this is a property of a circuit, a virtual circuit connects the two terminals. It can be a permanent virtual circuit or a virtual call.
virtual directory	Temporary directory created by the NFS client that is closer to the root in the file structure than the mount point. The virtual directory disappears once you dismount a filesystem.
virtual network	Network in which all connected hosts are able to communicate to each other as if they were all on the same local network. Users view an internet as a virtual network.
Virtual Terminal Protocol	*See* TELNET.

VMSINSTAL	OpenVMS installation procedure used to install TCPware products.
well-known port	Any of a set of protocol port numbers pre-assigned for specific uses by transport level protocols (TCP and UDP). Servers follow the well-know port assignments so clients can locate them. Examples of well-known port numbers include ports assigned to the remote login (TELNET) service and the file transfer (FTP) servers.
whitespace	Space, tab, or newline character.
WHOIS	Utility that allows Internet users to query the Network Information Center (NIC) username directory services.
wide area network (WAN)	Network element of an internet in which hosts connect over large geographic distances.
X.25	Set of networking recommendations that define the network/user interface in a Packet Switching Data Network (PSDN). X.25 provides a common set of protocols for computer systems to follow when interconnecting over a PSDN.
XQP	Extended QIO processor. *See* ancillary control process (ACP).