DEC TCP/IP Services for OpenVMS

Release Notes

April 1995

These Release Notes are for installers, managers, and end-users of:

- DEC TCP/IP Services for OpenVMS VAX Version 3.3
- DEC TCP/IP Services for OpenVMS AXP Version 3.3

This document updates the UCX Version 3.2 documentation set.

Operating System:	VMS VAX Version 5.5 OpenVMS VAX Versions 6.0, 6.1, 6.2 OpenVMS AXP Versions 6.1 and 6.2
Software Version:	DEC TCP/IP Services for OpenVMS VAX Version 3.3 DEC TCP/IP Services for OpenVMS AXP Version 3.3

Digital Equipment Corporation
Maynard, Massachusetts

Contents

2 Installation and Configuration

3 Corrected Problems

4 Known Problems

5 Restrictions

6 Documentation

7 Unsupported Components

A Installed Files

Examples

Figures

Tables

Preface

The DEC TCP/IP Services for OpenVMS (UCX) product is Digital's implementation of the TCP/IP protocol suite and internet services for OpenVMS VAX and OpenVMS AXP systems.

These Release Notes apply to:

- DEC TCP/IP Services for OpenVMS VAX Version 3.3
- DEC TCP/IP Services for OpenVMS AXP Version 3.3

Audience

This document is for UCX installers, managers, and end-users.

It is assumed that you are familiar with the UCX product and the Version 3.2 documentation set.

About This Manual

This document contains the following information:

- Chapter 1, "New Features"

  Lists the features new to this release and describes how to use and manage them.
- Chapter 2, "Installation and Configuration"

  Updates the Version 3.2 of the *DEC TCP/IP Services for OpenVMS Installation and Configuration* manual with information specific to UCX Version 3.3.
- Chapter 3, "Corrected Problems"

  Lists the problems in previous releases that have been fixed in Version 3.3.
- Chapter 4, "Known Problems"

  Describes the problems identified in the Version 3.3 software.
- Chapter 5, "Restrictions"

  Lists use restrictions of the software.
- Chapter 6, "Documentation"

  Updates the text of the Version 3.2 documentation.
- Chapter 7, "Unsupported Components"

  Documents the new software that is included with the Version 3.3 release but is not, in any way, legal or otherwise, supported by Digital.

Documentation Set

For information about UCX and how to use it, please use the Version 3.2 documentation set, which consists of the following manuals:

- *DEC TCP/IP Services for OpenVMS Concepts and Planning*

  Order Number AA–Q06TC–TE
- *DEC TCP/IP Services for OpenVMS Installation and Configuration*

  Order Number AA–LU49G–TE
- *DEC TCP/IP Services for OpenVMS User's Guide*

  Order Number AA–PC27F–TE
- *DEC TCP/IP Services for OpenVMS Management*

  Order Number AA–LU50G–TE
- *DEC TCP/IP Services for OpenVMS Management Command Reference*

  Order Number AA-PQQGD-TE
- *DEC TCP/IP Services for OpenVMS Sun RPC Programming*

  Order Number AA–Q06VC–TE
- *DEC TCP/IP Services for OpenVMS System Service and C Socket Programming*

  Order Number AA–LU51G–TE

1

New Features and Enhancements

The Version 3.3 release of the DEC TCP/IP Services for OpenVMS (UCX) product offers the following new software:

- Management control program (enhanced) (Section 1.1)
- NFS Client (enhanced) (Section 1.2)
- NTP (Section 1.3)
- RLOGIN and RSH (enhanced) (Section 1.4 and Section 7.3)
- SLIP and CSLIP (Section 1.5)
- TELNET (enhanced) (Section 1.6)
- UCX$TRACE (enhanced) (Section 1.7)
- LPD (enhanced) (Section 3.8)
- SMTP (enhanced) (Section 3.14)
- TELNET Print Symbiont (TELNETSYM) (enhanced) (Section 3.15.1)
- Outbound TELNET devices (unsupported) (Section 7.1)
- Remote Copy (RCP) (unsupported) (Section 7.2)

1.1 Management Control Program

The following features are new:

- Terminating TCP connections on demand

  With the UCX DISCONNECT DEVICE_SOCKET command, you can interactively terminate existing TCP connections, for example:

  ```
  UCX> DISCONNECT DEVICE_SOCKET BG123
  ```

- Managing SLIP

  See Section 1.5.5.

1.2 NFS Client

The Network File System (NFS) Client software now functions in two modes:

- OpenVMS-to-UNIX (Section 1.2.1)
- OpenVMS-to-OpenVMS (Section 1.2.2)

1.2.1 OpenVMS-to-UNIX Mode

No change from Version 3.2.

1.2.2 OpenVMS-to-OpenVMS Mode

If the NFS server node is running UCX Version 3.3, this mode is used automatically, except if you mount a container file system. If it is running any earlier version of the UCX server or any server other than UCX, the client runs in OpenVMS-to-UNIX mode.

When you upgrade the NFS Server software to Version 3.3, Version 3.3 client nodes must remount the file systems served by the upgraded server before they start using OpenVMS-to-OpenVMS mode.

For more information, see Section 6.2.2.8.

1.2.2.1 File Organizations and Record Formats

All OpenVMS file organizations and record formats are supported for both read and write access.

1.2.2.2 File Name Conversion

In OpenVMS-to-OpenVMS mode, file names appear in their unconverted form.

1.2.2.3 Export Options

If you add a file system entry to the Export Database specifying the PURGE_VERSIONS or DATA_CONVERSION options, the NFS Client in OpenVMS-to-OpenVMS mode ignores these options.

For example, if you issue:

```
UCX> ADD EXPORT "/path/name" /HOST=host /OPTIONS=DATA_CONVERSION
```

sequential files that are not STREAM_LF record format are converted to STREAM_LF for UNIX clients, but not for OpenVMS clients in OpenVMS-to-OpenVMS mode.

1.3 NTP

This release of UCX implements the Network Time Protocol (NTP), described in Request for Comments (RFC):

- 1059, "Network Time Protocol (Version 1), Specification and Implementation."
- RFC 1119 (Version 2)
- RFC 1305 (Version 3)

NTP synchronizes timekeeping and provides accurate time stamps. Hosts running NTP can participate in a TCP/IP network with fully synchronized time.

1.3.1 Strata Numbers

Strata numbers identify the participating peers:

- Lower-strata peers act as time servers.
- Higher-strata peers are clients.
- Hosts in both directions — time provider and time requester — are known as **peers**.
- An Internet Time Server, with the lowest number, Stratum 1, has a radio-clock-generated clock based on Universal Coordinated Time (UTC). NTP times are an offset of UTC, formerly Greenwich Mean Time (GMT).
- Any higher stratum host is, by definition, a Stratum 2 host or higher. Most participating hosts have a stratum higher than Stratum 2. They act as timekeepers for other hosts on the NTP network.
- Each host has its identifying stratum number encoded within UDP datagrams that these peers exchange periodically with each other. The datagrams are time-stamped. Each host adjusts its clock as needed.

1.3.2 Logging

NTP maintains a log file and reopens it daily, each time creating a new version of the file. The older versions are not automatically purged. Figure 1–1 shows a small portion of an NTP log file.

Figure 1–1 Example Section from an NTP Log File

```
* lost contact with peer 16.1.0.2
* sync source lost, free running
* re-acquired peer 16.20.208.8
* selected new sync source 16.20.208.8, now at stratum 5
 17:09:56.09    +0.004304 seconds, trip 0.1039, aperture 0.0143, hold 4
 17:10:06.09    +0.003959 seconds, trip 0.1039, aperture 0.0140, hold 3
 17:10:38.09    +0.004832 seconds, trip 0.1000, aperture 0.0148, hold 2
 17:11:10.09    +0.004043 seconds, trip 0.1000, aperture 0.0140, hold 1
 17:11:42.09    +0.002963 seconds, trip 0.1000, aperture 0.0130, hold 0
 17:12:14.17    +0.003922 seconds, new tick 100004 for 31 seconds
 17:12:46.18    +0.002536 seconds, new tick 100004 for 36 seconds
 17:13:50.25    +0.006192 seconds, new tick 100004 for 89 seconds
 17:15:58.18    +0.000944 seconds, new tick 100004 for 13 seconds
 17:20:14.28    +0.000444 seconds, new tick 100004 for 6 seconds
 17:24:30.38    +0.003322 seconds, new tick 100004 for 47 seconds
 17:33:02.29    +0.001390 seconds, new tick 100004 for 20 seconds
 17:50:06.40    +0.001011 seconds, new tick 100004 for 14 seconds
 18:24:14.41    -0.001323 seconds, new tick 100002 for 19 seconds
 18:58:22.42    -0.000461 seconds, new tick 100002 for 6 seconds
* re-acquired peer 16.1.0.2
* selected new sync source 16.1.0.2, now at stratum 3
 19:26:00.37    -0.005816 seconds, trip 0.2600, aperture 0.0158, hold 4
 19:26:10.49    -0.006081 seconds, trip 0.2200, aperture 0.0161, hold 3
 19:26:42.50    -0.012000 seconds, trip 0.2300, aperture 0.0120, hold 2
 19:27:14.54    +0.017324 seconds, trip 0.2700, aperture 0.0120, hold 1
 19:27:46.51    +0.001404 seconds, trip 0.2200, aperture 0.0114, hold 0
 19:33:38.66    -0.002879 seconds, new tick 100002 for 31 seconds
 19:38:10.76    -0.004792 seconds, new tick 100001 for 31 seconds
 19:45:06.97    +0.055078 seconds, discarded, trip > 0.132004
 19:48:02.92    -0.004358 seconds, new tick 100001 for 31 seconds
```

1.3.3 Setting Up NTP

Like the other services, you can configure NTP during the UCX configuration procedure. However, before you run the UCX startup procedure, you might first need to edit the NTP configuration file. Setting up NTP with this sequence of steps makes unnecessary the need to execute manually UCX$NTPD_STARTUP.COM. When NTP is configured, the UCX startup procedure automatically starts NTP.

To configure NTP, following these steps:

1. Run the UCX configuration procedure, SYS$MANAGER:UCX$CONFIG.COM.
2. From the Main Menu, select the SERVER Components Configuration Menu.
3. From the SERVER Components Configuration Menu, select NTP, which is Option 11 (see Section 2.2).
4. Either finish running the UCX configuration procedure or exit at this point.
5. Populate the NTP Configuration File (see Section 1.3.3.1).

6. If you started UCX before properly and fully configuring NTP, run the NTP startup procedure, SYS$MANAGER:UCX$NTPD_STARTUP.COM.

1.3.3.1 NTP Configuration File

Each host maintains an NTP configuration file with the names of participating peers. During the UCX configuration procedure, if you select to configure NTP, the procedure creates the example NTP configuration file SYS$SPECIFIC:[UCX$NTP]UCX$NTP.TEMPLATE.

Use a copy of this template to list the names of participating peers. Follow these steps:

1. Identify the peers you want to include in the configuration file. These are hosts with which you regularly exchange data and where accurate time coordination is important. Each host can be both a:
   - Time server, from which the local host synchronizes its time.
   - Higher-stratum host, which synchronizes with the local host.
2. Edit SYS$SPECIFIC:[UCX$NTP]UCX$NTP.TEMPLATE to add peer entries.
   - Format of each entry — Type the word "peer" (which is not case-sensitive) followed by the peer host's IP address.
   - Comment lines — If you want to add comment lines, begin each with a semicolon (;).
3. You do not need to identify the stratum of each peer. NTP determines this by the information it sends through the NTP packets.
4. To achieve redundancy, list as peers multiple NTP servers (systems at a lower stratum).
5. Save the file as SYS$SPECIFIC:[UCX$NTP]UCX$NTP.CONF.

   NTP periodically reads the last version of this file so that your changes to it are dynamic.

Figure 1–2 shows the template file.

Figure 1–2 Example NTP Configuration File

```
;       Copyright (c) Digital Equipment Corporation, 1995
;                        Example NTP Configuration File
; This template needs to be renamed to ucx$ntp.conf.
; The Network Time Protocol (NTP) provides synchronized timekeeping among
; a set of distributed time servers and clients. The local OpenVMS host
; maintains an NTP configuration file, UCX$NTP.CONF, of participating peers.
; UCX$NTP.CONF is maintained in the UCX$NTP sys$specific directory.
; The system manager populating this file determines with which peer hosts
; the local hosts should negotiate and synchronize. These are hosts with which
; you regularly exchange data and where accurate time coordination is an
; important factor.  Include at least one (but preferably two) peer hosts
; that you are assured:
; * provide accurate time
; * synchronize to Internet Time Servers (if they are not themselves
;   Internet Time Servers).
; Two hosts provide reliability in case one of the two hosts goes down.
; The NTP peer configuration file is dynamic and does not require a restart
; after being edited.
; Samples entries follow.  Replace them with your own peers. The order of
; the items is irrelevant. Each line  not commented with a ";" or "!" must
; begin with the word peer and be followed by the peer host's internet address.
peer 18.72.0.3
peer 130.43.2.2
peer 16.1.0.22
```

1.4 RLOGIN and RSH

The RLOGIN and RSH services have the following new enhancements.

1.4.1 Default Case

As of Version 3.2, the RSH and RLOGIN end-user commands have a new default case for the /LOWERCASE qualifier. /LOWERCASE now defaults to ON. This new default might affect both the UCX manager and end-user.

In the Proxy Database, if the user name is in:

- Uppercase — the user must use the /NOLOWERCASE qualifier.
- Lowercase — RSH and RLOGIN default to /LOWERCASE.

As of Version 3.3, the RLOGIN and RSH case-sensitivity flag defaults to CASE_INSENSITIVE. With this setting, the server accepts both the old and new default behaviors, that is, both all-uppercase and all-lowercase user names.

To set the case-sensitivity flag, issue:

```
UCX> SET SERVICE RLOGIN /FLAGS=(CASE_INSENSITIVE)
UCX> SET SERVICE RSH /FLAGS=(CASE_INSENSITIVE)
```

1.4.2 Incoming RLOGIN Proxy

See Section 7.3.

1.5 SLIP and CSLIP

This release of UCX implements the SLIP (Serial Line IP) and Compressed SLIP (CSLIP) protocols, described in RFC 1055, "A Nonstandard for Transmission of IP Datagrams Over Serial Lines: SLIP."

SLIP is a TCP/IP protocol for point-to-point serial connections. Use SLIP to connect across a serial line, rather than an Ethernet, FDDI, or Token Ring LAN. Each end of the link requires SLIP software. At least one end requires UCX.

Configure dedicated SLIP lines either during the UCX configuration procedure or interactively. You can use any standard OpenVMS terminal device as a SLIP line.

1.5.1 Features of SLIP

SLIP is a packet-framing protocol that defines a sequence of characters that frame IP packets on a serial line. SLIP:

- Is a reliable way to use serial lines to connect TCP/IP hosts and routers.
- Allows mixes of hosts and routers to communicate. Host-to-host, host-to-router, and router-to-router are common SLIP network configurations.
- Allows you to run only one protocol (TCP/IP) over a SLIP connection.
- Provides no addressing, packet type identification, error detection, error correction, or compression mechanisms.

1.5.2 Features of CSLIP

CSLIP provides header compression, especially beneficial for small packets, as described in Request for Comments (RFC) 1144, "Compressing TCP/IP Headers for Low-Speed Serial Links."

Header compression improves packet throughput.

1.5.3 Protocol Description

The SLIP Protocol defines two characters:

- END — hexadecimal C0 (decimal 192)
- ESC — hexadecimal DB (decimal 219)

To send a packet, a SLIP host starts sending the data in the packet. If a data byte is the same code as the END character, a two-byte sequence of ESC and hexadecimal DC (decimal 220) is sent instead. If the data byte is the same as an ESC character, a 2-byte sequence of ESC and octal hexadecimal DD (decimal 221) is sent instead. When the last byte in the packet has been sent, an END character is then transmitted.

There is no "standard" SLIP specification, nor a "defined" maximum packet size for SLIP. It is probably best to accept the maximum packet size used by the Berkeley UNIX SLIP drivers: 1006 bytes including the IP and transport protocol headers (not including the framing characters).

UCX's implementation of SLIP accepts 1006-byte datagrams and does not send more than 1006 bytes in a datagram.

1.5.4 Uses for SLIP

SLIP routes traffic over a serial line. SLIP has the following uses:

- Dedicated, or hard-wired, serial lines
- Dialups

You can use dial-up SLIP with these configurations:

- UCX host originates the dial-up connection.
- UCX host answers calls and establishes remotely initiated SLIP connections.

Figure 1–3 shows an example of a network configuration using SLIP. In this configuration, SLIP Network 3.0.0.0 connects Networks 1.0.0.0 and 2.0.0.0, which might be Ethernet, FDDI, Token Ring, satellite, or wireless networks. The gateways configured to make the connection are each multihomed:

- `gateway.a`'s connection to:
  - Network 1.0.0.0 — is its internet interface with IP address 1.0.0.25
  - Network 3.0.0.0 (SLIP) — is its internet interface with IP address 3.0.0.8
- `gateway.b`'s connection to:
  - Network 2.0.0.0 — is its internet interface with IP address 2.0.7.8
  - Network 3.0.0.0 (SLIP) — is its internet interface with IP address 3.0.9.9

Figure 1–3 Configuration of Distant Networks Connected by SLIP

```
Network 1.0.0.0                                    Network 2.0.0.0
|                                                             |
| *----------*                                  *----------*  |
|=| 1.0.0.24 |                                  | 2.0.0.16 |= |
| *----------*                                  *----------*  |
|                                                             |
|                                                             |
|                 ..............................              |
|                 .            SLIP            .              |
|                 .       Network 3.0.0.0      .              |
|                 .                            .              |
|                 .                            .              |
|                 .=3.0.0.8           3.0.0.9=.               |
|                ggggggggggggggg   ggggggggggggggg            |
|    1.0.0.25    g             g   g             g    2.0.7.8 |
|============== g  gateway.a  g   g  gateway.b  g ==============|
|                g             g   g             g            |
|                ggggggggggggggg   ggggggggggggggg            |
|                                                             |
|                                                             |
| *----------*                                  *----------*  |
|=| 1.0.0.26 |                                  | 2.0.0.18 |= |
| *----------*                                  *----------*  |
|                                                             |
```

1.5.5 Setting Up a SLIP Interface: Overview

Before you can communicate across a SLIP line, complete these prerequisite tasks:

1. Choose an IP address for the interface at each end of the line.

   Every network interface requires a unique IP address. For example, a host with an Ethernet interface, an FDDI interface, and three SLIP interfaces has five different IP addresses. For another example, see Figure 1–3, which shows `gateway.a` and `gateway.b`, each with an IP address for their Ethernet interface and a second IP address for their SLIP interface.

2. Establish a physical connection.

3. On the UCX host, configure the SLIP interface.

After you complete these tasks, set up the SLIP interface. Use these UCX management commands:

- SET INTERFACE — takes effect immediately. Stays in effect until the next UCX shutdown.
- SET CONFIGURATION INTERFACE — makes the change part of the permanent configuration. Takes effect upon the next UCX startup.

Table 1–1 shows the qualifiers of the SET INTERFACE and SET CONFIGURATION INTERFACE commands for managing SLIP interfaces.

Table 1–1 Configuring SLIP: SET INTERFACE and SET CONFIGURATION INTERFACE Commands

Qualifier	Description
/[NO]AUTO_START	Optional. Default: /AUTO_START. Automatically creates the interface upon UCX startup.
/COMPRESS= { ON, OFF, AUTOMATIC }	Optional. Default: No compression. Enables or disables TCP header compression (CSLIP). With /COMPRESS=AUTOMATIC, compression remains off unless the remote end begins to use it.
/FLOWCONTROL	Optional. Default: No flow control. Enables the special handling of XON and XOFF characters to work properly with modems that are configured to interpret these characters locally. Specify /FLOWCONTROL only if the host at the other end of the line is another UCX host. If you cannot use /FLOWCONTROL, configure your modem to pass all the XON and XOFF characters through transparently.
/HOST=*host*	Required. Local host name or IP address using the interface. If your host is multihomed, specify an address.
/SERIAL_DEVICE=*device*	Required. Identifies the interface as a SLIP device. Specify an arbitrary terminal device name. (Unlike Ethernet, FDDI, and Token Ring interface names, a SLIP interface name is not related to the OpenVMS device name.)

Example: The following command configures SLIP interface SL5, using the local IP address assigned to host CROW, with a subnet mask of 255.255.255.0. The interface uses the terminal device TTA3:. /COMPRESS enables TCP header compression (CSLIP). /FLOWCONTROL enables special handling of XON and XOFF characters.

```
UCX> SET INTERFACE SL5 /HOST=CROW /NETWORK_MASK=255.255.255.0 -
_UCX> /SERIAL_DEVICE=TTA3 /COMPRESS=ON /FLOWCONTROL
```

1.5.5.1 Setting Up SLIP: Hard-wired Lines

Follow these steps:

1. Establish a physical connection. Plug in a serial cable between the two host systems or ensure that they are both cabled to opposite ends of a leased line.
2. Decide on the IP address. Follow these guidelines:
   - When you select IP addresses, it is often helpful to place each physical link in a separate network, or subnet. Just as each Ethernet or FDDI LAN is typically assigned one subnet number, assign a separate subnet to each SLIP line.

     The interfaces at the two ends of the line would both be in that particular subnet.
   - If it is advantageous for you to assign to the remote host an address in the same subnet as your site LAN, use the proxy ARP feature (see Section 1.5.5.4).
3. Configure the SLIP interface. Issue the SET INTERFACE command, the SET CONFIGURATION INTERFACE command, or both. The /HOST and /SERIAL_DEVICE qualifiers are required.

   Example: This command defines interface SL2 and identifies it as SLIP device TTA1: with IP address 1.2.3.4. Communications on this line will use CSLIP.

   ```
   UCX> SET INTERFACE SL2 /HOST=1.2.3.4 /NETWORK_MASK=255.255.255.0 -
   _UCX> /SERIAL_DEVICE=TTA1: /COMPRESS=ON
   ```

1.5.5.2 Setting Up SLIP: Dial-up Lines

Either a terminal server port or an OpenVMS system itself can be configured to answer dial-in calls.

Follow these steps:

1. Configure the appropriate settings for the terminal port to which you will connect. Begin a dialog of dialing (or answering) commands with your modem. The specific required commands depend on the type of modem you are using.

   Example 1: The following command prevents the modem from hanging up when you exit the SET HOST /DTE session to bring up the SLIP line:

   ```
   $ SET TERMINAL TTA2 /PERMANENT /MODEM /NOHANGUP
   ```

   Example 2: This command disables interactive logins on the line. Any SLIP data that arrives before you issue the SET INTERFACE command is ignored. Otherwise, it would trigger the creation of a new interactive login process.

   ```
   $ SET TERMINAL TTA2 /PERMANENT /NOTYPEAHEAD
   ```

   Example 3: To allow interactive logins after a user sends a Break, issue this command:

   ```
   $ SET TERMINAL TTA2 /PERMANENT /NOAUTOBAUD /SECURE_SERVER
   ```
2. Configure the modem. Issue the appropriate commands to dial the telephone and establish communication.

3. Unless you are setting up a SLIP line between two UCX hosts and plan to use the /FLOWCONTROL qualifier at both ends, disable modem recognition of XON and XOFF characters. (If SLIP packets have CTRL/S and CTRL/Q characters embedded in them as data, you must prevent the modem from trying to interpret them.)

   Either use hardware flow control or disable flow control entirely. The following examples disable all flow control.

   Example 1: With a DECmodem V32 in AT command mode, set:

   `AT%F0` - No speed buffering flow control
   `AT%M0` - Disable speed buffering (optional)

   Example 2: With a DECmodem V32 in DCML mode, set:

   `SET P2/SBU and SET P1/SBU` - Then answer prompts

   Example 3: With a U.S.Robotics Sportster modem, set:

   `AT&B0` — Variable, follows connection rate (optional)
   `AT&H0` — Flow control disabled
   `AT&I0` — Software flow control disabled

4. Decide on the IP address. Follow these guidelines:

   – When you select IP addresses, it is often helpful to place each physical link in a separate network, or subnet. Just as each Ethernet or FDDI LAN is typically assigned one subnet number, assign a separate subnet to each SLIP line.

     The interfaces at the two ends of the line would both be in that particular subnet.

   – If it is advantageous for you to assign to the remote host an address in the same subnet as your site LAN, use the proxy ARP feature (see Section 1.5.5.4).

5. Dial in. Follow these steps:

   a. Issue the SET HOST /DTE command:

   ```
   $ SET HOST /DTE terminal_name
   ```

   b. Type:

   ```
   ATDT telephone_number
   ```

   c. The connected system displays its interactive (command mode) prompt. You are "talking" to the terminal server and can now make the connection.

Example: This example shows a user named SLIP-USER, at her PC named ROBIN and her 9600-baud modem, using terminal device TTA2 and connecting it to the port of terminal server. In this example:

- The terminal server is a DECserver 700 terminal server.
- It dials the telephone number 222-2222.
- The "password" prompt of the terminal server is `#`.
- Its current log-in password is `hootowl`.
- Its prompt is `Local>`.

- The user's user name is SLIP-USER.

```
$ SET HOST /DTE TTA2

%REM-I-TOQUIT, connection established
Press Ctrl/\ to quit, Ctrl/@ for command mode

ATDT 2222222

CONNECT 9600

# hootowl (not echoed)

Network Access SW V1.5 for DS700-16
(c) Copyright 1994, Digital Equipment Corporation - All Rights Reserved
Please type HELP if you need assistance

Enter username> SLIP-USER

Local> CONNECT SLIP
Ctrl/\
```

1.5.5.3 Setting Up SLIP: UCX As a Dial-up SLIP Provider

You can configure a UCX host to answer calls and establish remotely initiated SLIP connections. To set up this kind of connection:

1. Log in to the system.
2. Issue an appropriate UCX SET INTERFACE command.

   Specify the /SERIAL_DEVICE qualifier.

   Example: This command creates a SLIP interface named SL5, using the terminal device associated with the session where the command is issued.

   ```
   UCX> SET INTERFACE SL5 /HOST=192.208.35.5 /SERIAL_DEVICE=TT
   ```
3. Log out

As soon as you log out, UCX takes over your terminal port as a SLIP interface. Without causing the modem to hang up, start SLIP on the remote system.

To facilitate connection setup for end-users, create a dedicated user name for each remote host that dials in. These users need have a LOGIN.COM procedure that issues appropriate SET TERMINAL and UCX SET INTERFACE commands, terminating with a LOGOUT command. Every user should specify a different SLIP interface name and host name (or IP address). These users require the OPER privilege in order to create interfaces.

Optionally, enable IP forwarding on the SLIP provider host and start dynamic routing. As appropriate, issue:

```
UCX> SET PROTOCOL IP /FORWARD

UCX> SET CONFIGURATION PROTOCOL IP /FORWARD

UCX> START ROUTING /SUPPLY

UCX> SET CONFIGURATION START ROUTING /SUPPLY
```

IP forwarding must be enabled if the host is expected to act as a gateway, forwarding packets between the SLIP interface and other available interfaces.

Dynamic routing, with the /SUPPLY option, can be used to automatically send notifications on all connected LANs when new hosts or networks become reachable. For example, every time a SLIP connection is set up to a new remote subnet, a corresponding route will be advertised using RIP (the Routing Information Protocol).

1.5.5.4 Setting Up a Host "Directly Connected" to the LAN

You can make a SLIP-connected host appear as if it were directly connected to the LAN. This is possible using a proxy ARP server — usually the same host that is acting as a SLIP gateway into the LAN.

To use proxy ARP, assign to the remote host an IP address in the same subnet as the LAN. As other hosts on the LAN attempt to communicate with the remote host, the SLIP gateway answers ARP queries for the remote host by giving its own LAN address. The gateway then forwards packets across the SLIP line.

Many of Digital's DECserver terminal server products support SLIP connections and implement proxy ARP. If you dial in from a UCX host to a terminal server, it automatically detects what IP address you are using and begins responding to ARP queries, forwarding packets as necessary.

To use proxy ARP with a DECserver terminal server, assign an IP address in the same subnet as the terminal server.

On the terminal server, issue the SHOW PORT SLIP command. Verify that:

- A particular IP address has not already been associated with your port.
- Header compression is available, if you plan to use it.

1.5.5.5 UCX As a SLIP Gateway with Proxy ARP

It is also possible to set up a host running UCX as a SLIP gateway with proxy ARP. You might prefer this approach if your dial-in modems are attached directly to an OpenVMS system, rather than to a terminal server.

Follow these steps:

1. On the host to become a SLIP gateway, create a SLIP interface in another network or subnet, for example:

   ```
   UCX> SET INTERFACE SL0 /HOST=10.1.2.3 /SERIAL_DEVICE=TTA2
   ```

2. On the host to become a SLIP gateway, add a host route for the remote system, for example:

   ```
   UCX> SET ROUTE FINCH /GATEWAY=10.1.2.3
   ```

3. On the host to become a SLIP gateway, configure an ARP entry for the remote host, listing your own Ethernet address (as shown in UCX SHOW INTERFACE /FULL), for example:

   ```
   UCX> SET ARP 08-00-2B-2C-4F-46 FINCH /PUBLIC
   ```

4. On the host to become a SLIP gateway, enable IP packet forwarding, if not already done. Issue:

   ```
   UCX> SET PROTOCOL IP /FORWARD
   ```

5. On the remote host at the other end of the serial line, create an interface. Specify an address in the same subnet as the LAN.

Although the two ends of the SLIP line are in different subnets, traffic can flow properly due to the interface route you added with the SET ROUTE command.

1.5.6 Shutting Down SLIP

To terminate a SLIP connection, follow these steps:

1. Return the associated terminal port to general use. Issue:

   ```
   UCX> SET NOINTERFACE interface
   ```

2. If you added special route and proxy entries in conjunction with the SLIP line, remove them.

3. If you changed any terminal settings in preparation for SLIP, restore them. Issue the DCL SET TERMINAL command.

1.5.7 Troubleshooting

To debug SLIP problems, use the following methods:

- From another window, issue a TCPIPTRACE command to see packets going in and out of the system.
- Watch the modem's Send and Receive data LEDs as you attempt communication with the TELNET or PING commands.
- Display a count of the packets being sent and received on the problem interface, in full-screen format, updated every second. Issue:

  ```
  UCX> SHOW INTERFACE SLn /CONTINUOUS=1
  ```

1.6 TELNET

The TELNET Service offers the following new software:

- SHOW SESSION command (Section 1.6.1)
- Command qualifiers (Section 1.6.2)
- Error and status messages (Section 1.6.3)

1.6.1 SHOW SESSION Command

The SHOW SESSION command is now implemented. If you are running TELNET with:

- No connections, the command displays:

```
TELNET> SHOW SESSION
%TELNET-E-NOSESSION, No active session
```

- One session, the command displays information such as:

```
TELNET> SHOW SESSION
Session 01, host ucxaxp, port 23  (default active session)
```

- Multiple sessions, the command displays information such as:

```
TELNET> SHOW SESSION
Session 01, host ucxaxp, port 23
Session 02, host tnpubs, port 23  (default active session)
```

1.6.2 Command Qualifiers

The TELNET and TN3270 commands have the new qualifiers shown in Table 1–2.

Table 1–2 TELNET and TN3270 Commands: New Qualifiers

Qualifier	Description
/NOINTERACTIVE	Optional. Default: TELNET command mode. Disables the capability of the escape character to leave the session and return you to the TELNET prompt (useful when the TELNET or TN3270 commands are issued from a command procedure in a captive account).
/LOG_FILE=*file*	Optional. Default: no logging. Logs the output of all sessions to the specified file while maintaining the sessions on your terminal.

1.6.3 Error and Status Messages

The following messages enhance the TELNET Service:

- `TELNET-S-NORMAL Normal successful completion`

  Successful connection.

- `TELNET-E-CONNFAIL Failed to connect to remote host`

  TELNET's attempt to connect to the remote host failed.

- `TELNET-E-CREATELOG Failed to create log file` *`file_name`*

You requested an output log file with the /LOG_FILE qualifier, but TELNET's attempt to create the file failed.

- `TELNET-I-ESCAPE Escape character is` *`char`*

  This is the default or user-defined escape character for your session.

- `TELNET-S-EXIT EXIT requested`

  You entered the EXIT command. TELNET exits.

- `TELNET-I-INETCALL` *`message`*

  Information about the interface call that triggered an error. This message accompanies the TELNET-E-INETERROR message.

- `TELNET-E-INETERROR Internet interface error`

  TELNET's attempt to issue an internet call failed. This message is usually accompanied by the TELNET-E-INETCALL message to show both the actual call that was attempted and the resulting status of the call.

- `TELNET-E-IVHOST Invalid or unknown host` *`host`*

  TELNET's attempt to resolve the specified host's address failed.

- `TELNET-E-IVMODE Invalid mode specified`

  You tried to set the mode to something other than CHAR or LINE. This is not valid.

- `TELNET-S-LCLCLOSED Local connection closed`

  You closed the TELNET connection by issuing either the DISCONNECT or EXIT command.

- `TELNET-E-NOLICENSE License check failed`

  The DEC TCP/IP Services for OpenVMS license is: not present, not valid, or disabled.

- `TELNET-E-NOSESSION No active session`

  You issued a command, which is valid during a session, but there is no current session or no sessions exist.

- `TELNET-E-NRESNOTAVAIL Network resource not available`

  TELNET tried to perform a network I/O, but there were no network buffers available.

- `TELNET-E-OPENIN Failed to open` *`file`* `for input`

  Attempt to access the TELNET initialization file, SYS$LOGIN:TELNETINIT.INI, resulted in an error. The next messages indicate the reason.

- `TELNET-E-OPENQUOTA Too many sessions open`

  You tried to create more than the maximum number of allowed simultaneous sessions, 64.

- `TELNET-S-REMCLOSED Remote connection closed`

  An active connection was closed by the server.

- `TELNET-S-RESUME Resume TELNET session`

  You issued a RESUME command to resume your session.

- `TELNET-I-SESSION Session` *`session_number`*, *`host`*, *`port_number`*

The session number, host name, and port number of the session whose status is being displayed.

- `TELNET-E-TERMERROR Terminal interface error`

  TELNET's attempt to issue a QIO failed.

- `TELNET-I-TERMFUNC` *`function_name`*

  The function that was being attempted when an error was returned to a QIO. This message follows the TELNET-E-TERMERROR message.

- `TELNET-I-TRYING Trying ...` *`host`*

  Host to which a connection is being attempted.

- `TELNET-W-UNSUPPORTED Qualifier` *`qualifier`* `is not currently supported`

  The qualifier you specified is not supported.

Note

In OpenVMS Version 6.2, the /AUTHENTICATE qualifier of these DCL commands is not supported: SET HOST /TELNET and SET HOST /TN3270.

1.7 UCX$TRACE

The UCX$TRACE Utility has the following new features:

- New DCL TCPIPTRACE command — Use this command to invoke the utility.
- New options — Use the qualifiers listed in Table 1–3 to customize tracing for your particular problem.

Table 1–3 TCPIPTRACE Command Qualifiers

Qualifier	Function
/BUFFERS=*n*	Optional. Default: 100. Number of buffers that UCX$TRACE allocates for temporary storage. These buffers must be locked into the working set, so the number can be: • Decreased to be accommodated in the working set • Raised to prevent the dropping of trace packets
/FULL	Optional. Default: brief display. Displays the packet's contents.

(continued on next page)

Table 1–3 (Cont.) TCPIPTRACE Command Qualifiers

Qualifier	Function
/OUTPUT=*file*	Optional. Default: Screen display. Redirects the output from screen to the specified file. If this file name already exists, the output is appended to it.
/PACKETS=*n*	Optional. Default: 10. Stops the trace after the specified number of packets is displayed.
/PORT= [LOCAL / REMOTE] =*n*	Optional for port number. Default: all traffic is displayed. Required for port type. Filters the trace to the specified port.
/PROTOCOL= { ARP / ICMP / IP / TCP / UDP }	Optional. Default: /PROTOCOL=IP. Filters on the specified protocol.

Examples:

```
$ TCPIPTRACE HOST1 /FULL /PORT=REMOTE=21
```

```
$ TCPIPTRACE HOST2 /PORT=(LOCAL=23, REMOTE=1056) -
_$ /FULL /PACKETS=30 /OUTPUT=TELNET_TRACE.TXT
```

2

Installation and Configuration

This chapter plus the *DEC TCP/IP Services for OpenVMS Installation and Configuration* manual provide complete installation information for Version 3.3.

2.1 Requirements

Check that your system meets all the requirements for installation.

2.1.1 Operating System

The DEC TCP/IP Services for OpenVMS (UCX) Version 3.3 software requires one of the following operating systems:

- VMS VAX Version 5.5
- OpenVMS VAX Version 6.1 or 6.2
- OpenVMS AXP Version 6.1 or 6.2

2.1.2 Licenses

To use UCX, you need one of these licenses:

- DEC TCP/IP Services for OpenVMS
- DEC TCP/IP Client for OpenVMS
- Digital NAS Client 150 Version 7.0 for OpenVMS VAX Workstations
- Digital NAS Client 150 Version 7.0 for OpenVMS APX Workstations
- Digital NAS 150 to 250 Upgrades

DECwindows — You do not need a UCX license to display DECwindows applications on a remote TCP/IP host.

PATHWORKS — You do need a license for loading the software supplied for PATHWORKS systems.

2.1.3 Information You Need

For a smooth and fast installation and configuration, before you start the procedures:

- Read these Release Notes.
- Read the *DEC TCP/IP Services for OpenVMS Installation and Configuration* manual.
- Complete the installation and configuration checklists with the required information (see the *DEC TCP/IP Services for OpenVMS Installation and Configuration* manual).

2.2 Server Configuration Menu

The Server Components Configuration Menu has been updated to add the Network Time Protocol (NTP) Service (for NTP information, see Section 1.3).

Figure 2–1 shows a sample TCP/IP Services for OpenVMS SERVER Components Configuration Menu.

Figure 2–1 Example: SERVER Components Configuration Menu

```
        TCP/IP Services for OpenVMS SERVER Components Configuration Menu

        Configuration options:

                 1  -  BIND           Disabled
                 2  -  BOOTP          Disabled
                 3  -  TFTP           Enabled
                 4  -  FTP            Enabled
                 5  -  LPR/LPD        Enabled
                 6  -  NFS            Enabled
                 7  -  PC-NFS         Enabled
                 8  -  PORTMAPPER     Enabled
                 9  -  TELNET/RLOGIN  Enabled
                10  -  SNMP           Disabled
                11  -  NTP            Disabled

                 A  -  Configure options 1 - 11
                [E] -  Exit menu

Enter configuration option:
```

2.3 New Images

DEC TCP/IP Services for OpenVMS Version 3.3 includes the following new files:

- SYS$COMMON:[SYSLIB]UCX$SDA.EXE
  - System Dump Analyzer (SDA) back end
  - Debugs problems within the PATHWORKS Internet Protocol (PWIP) Driver
- SYS$COMMON:[SYSEXE]UCX$RCP.EXE
  - RCP command
- SYS$COMMON:[SYSEXE]UCX$NTPD.EXE
  - Network Time Protocol (NTP)
  - Sends and receives network messages with the correct time and adjusts the system clock

DEC TCP/IP Services for OpenVMS Version 3.3 also updates the following help files:

- UCX$TOPLEVEL_HELP.HLP
- UCX$UCP_HELP.HLP
- UCX$TELNET_HELP.HLP

For a list of all the files in the distribution kit, see Appendix A.

2.4 Reinstalling

If your installation is an upgrade, or if you are reinstalling an older version of the UCX product, complete the following steps.

2.4.1 Upgrading from UCX Version 3.0 or Earlier

If you are upgrading from UCX Version 3.0 or earlier, reconfigure all the currently configured server and client components.

2.4.2 Reverting to a Previous Version

If you decide to revert to an earlier version of UCX, follow these steps:

1. Run the cleanup procedure, SYS$UPDATE:UCX$CLEANUP.COM, which deletes most UCX data. UCX$CLEANUP deletes UCX-related executables, directories, accounts, identifiers, and save sets. However, it does not delete database files.
2. Delete the Configuration Database, SYS$COMMON:[SYSEXE]UCX$CONFIGURATION.DAT.
3. Shut down UCX.
4. Reinstall the version you want to use.

2.4.3 RSH and REXEC

If you are upgrading from a previous release, move the Remote Shell and Remote Execute startup command files to their own directories on SYS$SYSDEVICE. Use the following commands:

```
$ COPY SYS$MANAGER:UCX$REXECD_STARTUP.COM SYS$SYSDEVICE:[UCX$REXEC]

$ COPY SYS$MANAGER:UCX$RSHD_STARTUP.COM SYS$SYSDEVICE:[UCX$RSH]
```

3 Corrected Problems

The following problems in previous releases of the software are fixed in Version 3.3:

- Kernel (Section 3.1)
- PWIP Driver (Section 3.2)
- Sun RPC programming interface (Section 3.3)
- Management control program (Section 3.4)
- BIND Server (Section 3.5)
- Remote BOOT Server (Section 3.6)
- FTP (Section 3.7)
- LPD (Section 3.8)
- NFS Client (Section 3.9)
- NFS Server (Section 3.10)
- PC-NFSd (Section 3.11)
- Portmapper (Section 3.12)
- RSH (Section 3.13)
- SMTP (Section 3.14)
- TELNET (Section 3.15)

3.1 Kernel

These "bugs" are fixed:

- Multiple open sockets

  UCX now supports the select( ) operation on more than 32 open sockets at the same time.
- The IO$_SETMODE function of the $QIO[W] system service no longer causes access violations or system crashes.
- INET device units are now correctly deleted on final deassignment.
- The parsing table for remote table allows an "ignore" bit.
- The configuration of communication security lists is now supported.
- UCX now correctly searches the proxy cache when wildcard proxies were involved. Dynamically loaded proxies now correctly appear to be loaded, and non-wildcard proxies are correctly found when similar wildcard proxies exist for different user names.
- The SHOW COMMUNICATION /MEMORY command no longer triggers a problem with corrupted pool.
- When two processes share a socket, only the actual creator could perform IO$_DEACCESS QIOs. UCX now checks these items:
  - Is the socket shareable?
  - Does the issuing process have the SHARE privilege?
  - Does the issuer's parent process own the device?

  Access is denied only if all checks fail.
- A problem with dynamic routing on OpenVMS AXP systems is fixed.
- A problem with the execution of the SHOW INTERFACE command is fixed.
- UCX correctly aborts FTP from remote multivendor hosts.
- UCX now correctly deallocates a VCRP.
- When starting a new server process, the Auxiliary Server now uses the LOGINOUT.EXE image rather than the DCL.EXE image.
- SELECT on a multiprocessor with null lists now works properly.

3.1.1 Security Driver

- The Security Driver works properly.
- The Security Driver interface is fixed.

3.1.2 TELNET (TN) Driver

- UCX processes correctly if a partner for a TN Driver connection becomes unreachable.
- A problem due to a corrupted TIMER queue is fixed.
- When the TELNET Server receives both a DO ECHO and a WILL ECHO from the remote host, UCX now correctly sets the remote echo flag.
- UCX now correctly limits the size of the subnegotiation buffer.

- The problem with a TN Driver UCB overrun into the following ORB is fixed.

3.1.3 UCX$TRACE Utility

- The utility properly counts only the number of packets displayed.
- To better distinguish between the end of one packet and the start of the next, each each packet now starts with 3 blank lines and one line of dashes (-).
- Processing is successful if the number of buffers is one.
- The formatting of the UDP destination port is now correct.

3.2 PWIP Driver

These "bugs" are fixed:

- PWIP shutdown and restart processes correctly.
- Link error processing is now handled in a manner consistent with the Transport Provider Interface (TPI) Specification.
- The PWIP Driver does not attempt to transmit if transmit is suspended.
- The PWIP ACP log file name, PWIPACP_*host*.LOG, is now UCX$PWIPACP_*host*.LOG.
- Records are now flushed to disk after every write. On OpenVMS AXP systems, UCX correctly reads the PWIP ACP log file.
- Attempts to assign channels to BG devices are now successful.
- Sessions queued to the PWIP Driver work queue are properly removed before termination processing proceeds.
- The PWIP Driver startup and shutdown messages are now posted to OPCOM.
- The PWIP Driver now dynamically recalculates MBUF resource and session limits when they are exceeded. This removes the need to reboot the system in order to pick up UCX large buffer maximum changes.
- The PWIP Driver now writes a message to OPCOM when MBUF resource and session limits are depleted or replenished.
- Backlog enforcement works correctly.
- A problem with untimely disconnect requests because of timing issues has been identified and fixed.
- For connect reject processing, the appropriate interface state is no longer overwritten with an inappropriate state in routine `d_discon_req` in module PWIPDRIVER_DOWN.C.
- Changes have been made to the MBUF transmit throttle so that PWIP has sufficient available buffers, particularly when a large number of sockets are active.
- UCX correctly defers the queuing of connections to the ACP work queue if the connection is already queued to the PWIP Driver work queue.

3.3 Sun RPC Programming Interface

These "bugs" are fixed:

- The definition and usage of the XDR_VECTOR( ) `addrp` argument is now correct.
- CLNT_SPERRNO( ), CLNT_SPCREATE_ERROR( ), and CLNT_SPERROR( ) now include the VAX transfer vectors and AXP symbol vectors.

3.4 Management Control Program

These "bugs" are fixed:

- SET COMMUNICATION /INITIALIZE command — now in the UCX startup procedure.
- 15-character IP addresses — now resolved correctly.
- UCX configuration procedure — no longer requires that you specify your host's domain in order for you start UCX.
- UCX startup procedure — no longer has problems with a local host name if either its canonical name or fully qualified name is a subset of the other.
- Message Exchange (MX) lookups — no longer cause a memory leak.
- The network management commands in Table 3–1 have been fixed.

Table 3–1 Corrected Problems with Management Commands

Command	Fix
SET NOBOOTP	If a host has duplicate IP addresses (2 records, 2 identical hardware addresses, 2 different IP addresses), the command now deletes the appropriate record.
SET CONFIGURATION NOBIND	No longer removes only the primary BIND server from the Configuration Database.
SET MX_RECORDS	In a host lookup, if UCX cannot find the host specified with the `destination` parameter, the software now correctly creates an MX record.
SET NOMX_RECORDS	An unknown destination no longer results in an error.
SET NOMX_RECORDS *	Wildcarding no longer causes access violations.
SET NOMX_RECORDS * /GATEWAY=*gateway*	The value of the /GATEWAY qualifier is no longer treated as a wildcard.

(continued on next page)

Table 3–1 (Cont.) Corrected Problems with Management Commands

Command	Fix
SET CONFIGURATION NAME_ SERVICE	When you specify multiple hosts as BIND servers, this command now properly maintains the order, as you enter it, of the server list.
CONVERT/ULTRIX BIND	Can create a reverse domain.
SET CONFIGURATION COMMUNICATION	Properly supports /REMOTE=NOVIRTUAL.
SHOW CONFIGURATION BIND /OUTPUT=*file*	Works correctly.
SHOW ROUTE /PERMANENT	Works correctly with alias names.

3.5 BIND Server

These BIND Server problems are fixed in the Version 3.3 software:

- The following VAX-only problem with the SHOW NAME_SERVICE /QUERIES and SHOW NAME_SERVICE /STATISTICS commands is fixed.
- A problem with buffer flow has been fixed.

3.6 Remote BOOT Server

BOOTP now provides optional vendor data for the boot file size and gateways.

3.7 FTP

These "bugs" are fixed:

- The messages associated with the following FTP commands now conform to RFC 959:
  - `mkd` (make directory)
  - `pwd` (show default)
- The DIRECTORY /BRIEF command now produces the same output as the UNIX `ls` command. The default is /FULL.
- Client and Server command-synchronization loss no longer occurs.
- The `mget` command supports full path name specifications.
- Binary files with a size not in multiples of 512 bytes are now corrected transferred over DECnet.
- The FTP command-line qualifiers /USER, /PASSWORD, and /INPUT now work properly.
- A remote PASV command no longer hangs up the session.
- The PUT and GET commands used with the /FDL qualifier now work properly.
- More specific reply codes improve Client and Server error reporting.
- The GET command appropriately uses only the file name, not the full path name.
- If VMS Plus mode is enabled, FTP properly does wildcard expansion.
- When you issue `mget`, FTP properly does wildcard expansion.
- Unless VMS Plus mode is enabled, OpenVMS-style GET transfers only one copy, and UNIX-style `mget` transfers only multiple copies.
- The GET command no longer strips device and directory information from the local file specification.
- Users no longer get the number of bytes transferred even if none were due to error.
- Writing to tape, FTP now satisfies tape driver size requirements.
- The GET, PUT, and DIRECTORY commands now work correctly if you specify the following syntax:

  `node::node::device:[directory]file.type`

  If you copy a directory, it becomes unreadable on the target end. If access is denied, the returned error is:

  ```
  Failed to open file
  ```
- FTP now correctly handles multidirectory linked logical names.
- The PUT command processes correctly with an even number of subdirectories.
- A search list correctly releases all channels.
- FTP now writes to nondirectory devices such as LTA: and terminals.

- The PUT command to a concealed device with DIRECTORY [000000] now works properly.

3.8 LPD

These "bugs" are fixed and the following enhancements supplied.

Note

This section includes new documentation.

- If mail notification is requested, LPD running in UCX$LPD_QUEUE now works correctly.
- Outbound PRINT /PASSALL of a file with an embedded carriage control no longer causes double-spaced printing. (The most common cause of this problem was the printing a non-stream LF file from an application ending its text lines with <LFCR> rather than <CRLF>.)

  If you have a problem with /PASSALL printing, you can revert to the old behavior. Issue:

  ```
  $ DEFINE /SYSTEM UCX$LPD_STREAM_PASSALL 1
  ```

  However, if you have a problem with /PASSALL printing that seems to be fixed by doing this, please report it to Digital.
- All LPD Receiver diagnostics are now written to UCX$LPD_RCV_STARTUP.LOG (UCX$LPD_RCV_LOGFILE.LOG is obsolete).
- The LPD Server no longer loses jobs submitted in quick succession from UNIX clients.
- LPD honors all OpenVMS FLAG PAGE print options, including:
  - /FLAG qualifier of the DCL PRINT command
  - /DEFAULT=FLAG setting on the LPD print queue
  - /SEPARATE=FLAG setting on the LPD print queue

  To turn on this behavior, define the system logical name UCX$LPD_VMS_FLAGPAGES. At queue start-up time, issue:

  ```
  $ DEFINE /SYSTEM UCX$LPD_VMS_FLAGPAGES 1
  ```

  When you define UCX$LPD_VMS_FLAGPAGES, LPD:
  - Honors what OpenVMS instructs regarding flag pages for outbound jobs.
  - Submits inbound jobs with /FLAG or /NOFLAG, based on the presence of the 'L' card in the control file. Inbound jobs with:
    - An 'L' card — are submitted to the destination print queue as PRINT /FLAG.
    - No 'L' card — are submitted to the destination print queue as PRINT /NOFLAG.
  - Renders meaningless the DCL PRINT /PARAM=NOFLAG command.

  If you do not define this logical name, LPD behaves as it did before.
- The ROOT account no longer requires a communication proxy in the Proxy Database regardless of the setting of the APPLICATION_PROXY flag.

- If the SMTP transport is not installed, the LPD symbiont correctly delivers a mail notification.
- If jobs are pending in a stopped LPD queue, you can now successfully restart the queue.
- For outbound print jobs, LPD now has a "retry" capability.
- To identify a PostScript™ file, LPD now supports both the "o" card or the "`Dpostscript`" card.

  Configuring LPD to use the LPD standard "o" card to indicate a PostScript file is useful for printing PostScript files over LPD to a system that does not implement the Digital LPD Printserver Extensions (PS Extensions).

  You can configure this option for individual queues with the new "ps" field in the Printcap File. Edit the Printcap entries, adding a "ps" field, with one of these values:

  - "non_PS" — indicates that the remote system does not implement PS Extensions.
  - "LPS" — Indicate that the remote system implements PS Extensions.

  You can configure this option systemwide with the new UCX$LPD_PS_EXT system logical name. The legal values for this logical are the same as for the 'ps' Printcap field — "non_PS" and "LPS." For each Printcap entry that does not have a 'ps' field defined, LPD uses the value you assigned to the UCX$LPD_PS_EXT logical name. If you did not define UCX$LPD_PS_EXT, LPD defaults to using the PS extensions.

  Configurations with no defined UCX$LPD_PS_EXT logical name and no "ps" fields in the Printcap file, continue to work exactly as they always have. No configuration changes are required by this LPD enhancement.

  The legal values are not case sensitive.
- Adding printer entries to the Printcap Database no longer causes the name of earlier entries to "bleed over" into the names of later ones, if the earlier name is longer than the later one.
- The PRINT /PARAMETER=PRINTER="*printer*" command now preserves the case of the name when you print to a UNIX system.
- Issuing STOP /QUEUE /RESET for an LPD queue while a job is processing no longer aborts the job.
- With inbound jobs, if the remote host from the 'H' control card is unknown, LPD no longer inappropriately retains the job in UCX$LPD_QUEUE.

  If the remote host from the 'H' control card is not known and the SET SERVICE LPD /FLAGS=APPLICATION_PROXY is:

  - Not set — the job goes through.
  - Set — UCX rejects the job.
- When a remote LPD client sends the control file before the data file, UCX no longer retains the job in the UCX$LPD_QUEUE queue with a "file not found" error.
- Outbound LPD jobs no longer fail with "File locked by another user."

- LPD proxy mapping now finds proxies when you wildcard /REMOTE_USER, for example:

```
UCX> ADD PROXY /REMOTE_USER="*" /HOST=host.name
```

- If SYS$SCRATCH is not defined in the system table and an inbound print request with a mail notification request comes in, the UCX$LPD_QUEUE symbiont now works properly.
- If LPD has a problem with opening SYS$SPECIFIC:[UCX_LPD]UCX$LPD_RCV_LOGFILE.LOG and the logical name to turn on the diagnostics (LPD_RCV) is not defined, LPD no longer sends diagnostics to the operator console.
- An inbound job completion OPCOM message is now generated if the LPD service log option LOGOUT is set.
- The LPD_RCV and LPD_DEBUG logical names for diagnostics are enhanced to track what the LPD Client sends compared to what the Receiver receives.

  If you have used the old LPD diagnostics by defining the system logical names LPD_RCV and LPD_DEBUG to 65535, the new value to get the same effect is 7. Therefore, replace:

  ```
  $ DEFINE /SYSTEM LPD_DEBUG 65535
  $ DEFINE /SYSTEM LPD_RCV 65535
  ```

  with

  ```
  $ DEFINE /SYSTEM LPD_DEBUG 7
  $ DEFINE /SYSTEM LPD_RCV 7
  ```

  LPD_DEBUG and LPD_RCV are bit-mapped values. The low-order three bits are used to give the effect previously produced by all the bits being set (old value of 65535).

  A new diagnostic is the fourth bit. If it is set (value 8), the LPD symbiont logs each buffer that it sends over the TCP/IP link, and the LPD Receiver logs each buffer that it receives from the TCP/IP link. This provides a way to see exactly what the LPD is sending (for outbound jobs) and receiving (for inbound jobs).

  To turn on just this new diagnostic and no other diagnostics, set the fourth bit of the LPD_DEBUG and LPD_RCV logical names by defining them to 8.

  To get the old diagnostics along with this new one, define LPD_DEBUG and LPD_RCV to 15 (recommended only for the diagnosis of problems). Leaving the diagnostics on during normal use slows the performance of LPD and produces large log files. LPD_DEBUG and LPD_RCV are independent:

LPD_ DEBUG	Applies to outbound jobs (LPD Client). Writes diagnostics to an LPD queue's log file.
LPD_RCV	Applies to inbound jobs (LPD Server). Writes diagnostics to the Receiver's log file, UCX$LPD_RCV_LOGFILE.LOG.

3.9 NFS Client

These Version 3.2 problems are fixed:

- When you issue the MOUNT command with a lowercase or mixed case volume label, the quotation marks no longer appear in the resulting volume label.
- When you issue the MOUNT command and specify a volume label, MOUNT qualifiers, including /PROCESSOR and /FILEIDS, are no longer ignored.
- The default ADFs now work correctly.

3.10 NFS Server

- NFS now correctly accesses files with any form of variable length records.
- If you configure ADD EXPORT /OPTIONS=PURGE_VERSIONS, the `ls` command to display NFS files now works correctly, even with the presence of a file with null name and extension, for example, .;1.
- The file system works corrctly when the final record of a file crosses the EOF mark.
- When you define the UCX$CFS_MODUS_OPERANDI mask 512 to enable REAL_SIZE determination, write operations work properly.

3.11 PC-NFSd

These problems are fixed:

- PC-NDSd no longer rejects logins to accounts with "password expired."
- Authentication now logs errors.
- When the client sends a fully qualified host name in its RPC request to print, PC-NFSd correctly creates the subdirectory.
- When PC-NFSd renames a print file into the spool subdirectory, and a file by the same name is already queued up for printing, PC-NFSd now works correctly.

3.12 Portmapper

The Portmapper now correctly writes error messages to the log file when registration fails.

3.13 Remote Shell

These problems are fixed:

- The RSH command now properly maintains exit status.
- RSH now executes properly on a nonterminal.

3.14 SMTP

The following "bugs" are fixed and enhancements supplied.

Note

This section includes new documentation.

- You can now reply to mail forwarded with the OpenVMS Mail SET FORWARD command.
- SMTP now handles addresses that include a source route, which, upon receipt, appear on the VMS Mail `From:` line, for example:

  `<@PSUVM.PSU.EDU:owner-acm-l@UNIVVM.UNIV.EDU>`
- SMTP now correctly truncates RFC 822 headers that exceed 1000 characters.
- Users no longer receive an error when trying to send a file with MAIL /NOEDIT and a DECnet address appears before an SMTP address, for example:

  `To: SOMNOD::JANE, SMTP%"john@someplace.else.org"`
- SMTP now delivers mail to an address with a dot in the local part of the address, for example: `To: G.Peach@xyz_corp.com`.
- SMTP no longer looks at a local-part of an address within quotation marks. An example is a local-part with a percent (%) character all within quotation marks, such as:

  `"PSI%abcdef.13911551526::somebody"@someplace.xyz_corp.com`
- SMTP handles messages even if the RFC header block has:
  - More than 100 lines of text
  - More than 8192 bytes
- SMTP correctly handles addresses with special characters inside a comment string, for example, " > " or " < ".
- The SMTP VRFY and EXPN commands work properly. (Some systems do a VRFY on a user or an EXPN on a list before sending mail.)
- SMTP now recognizes as "local" mail that is addressed to the substitute domain.

  (If you do not want it to recognize mail sent to the substitute domain as "local," define the system logical name UCX$SMTP_NO_SUBS_DOMAIN_INBOUND before you start the SMTP mail queue."
- Inbound mail
  - When mail comes in from a non-SMTP user to a user who has forwarded his mail via SMTP on the local system, the VMSmail `From:` line no longer rejects the REPLY command.

    If the non-SMTP transport from which the mail comes is DECnet, the `From:` line is no longer enclosed in quotation marks.

    If the non-SMTP transport is MTS, the `From:` line is no longer enclosed in quotation marks and no longer contains backslashes preceding the inner quotation marks in the address. **Examples:**

Old: `From: "SOMNOD::SOMEBODY"`

Corrected: `From: SOMNOD::SOMEBODY`

Old:

`From: "MRGATE::\"MRGATE::SUMNOD::SENDING_USER\""`

Corrected:

`From: MRGATE::"MRGATE::SUMNOD::SENDING_USER"`

- SMTP handles inbound print requests with mail notification requests.
- SMTP sets the VMSmail `CC:` line.
- If an X-VMS-To or X-VMS-Cc RFC header is present, SMTP puts the header in the VMSmail `To:` line.
- If an X-VMS-To or X-VMS-Cc RFC header is not present, SMTP puts the header in the VMSmail `To:` line.

  (If you do not want to put these headers in the VMSmail `To:` and `CC:` lines, define the system logical name UCX$SMTP_INBOUND_NOXVMS before SMTP execution queue startup.)
- SMTP handles multiline replies to the HELO command.
- SMTP handles forwarded incoming mail even if the SMTP configuration database relay option is not set.
- When forwarding incoming mail, SMTP no longer creates a new block of RFC headers, in addition to putting the old headers inside of the text of the new message.

3.15 TELNET

The following TELNET "bugs" are fixed.

3.15.1 TELNET Print Symbiont (TELNETSYM)

The TELNET Print Symbiont (TELNETSYM) software is corrected and enhanced as follows:

- Each TELNETSYM queue has its own log file. (The UCX$TELNETSYM_LOGFILE logical name is obsolete.)
- OPCOM messages sent by TELNETSYM now include the name of the execution queue.
- If you define UCX$TELNETSYM_SUPPRESS_FORMFEEDS, TELNETSYM no longer prints over the last line of text.
- TELNETSYM now correctly puts out Form Feeds (<FF>) at the end of each job.
- If you do not define the UCX$TELNETSYM_IDLE_TIMEOUT logical name, TELNETSYM no longer truncates the last blocks of print output.
- TELNETSYM sets its process names to UCX$TNSYM1, UCX$TNSYM2, UCX$TNSYM*n*.
- To configure printers that do not implement a TELNET layer, use the new system logical name UCX$TELNETSYM_RAW_TCP. Before you start the TELNET print queue, follow these steps: define UCX$TELNETSYM_RAW_TCP to 1.
- If you define the logical name UCX$TELNETSYM_LOG_KEEP, TELNETSYM purges log files. (If the logical name is not defined, no log purging occurs.) With purging, you can also use UCX$TELNETSYM_LOG_KEEP to set the maximum number of log files to keep.

  Example: This example enables purging after more than three copies of the same log file are created:

  ```
  $ DEFINE /SYSTEM UCX$TELNETSYM_LOG_KEEP 3
  ```
- To set the number of execution queues that a TELNETSYM process can serve, define the new TELNETSYM logical name UCX$TELNETSYM_STREAMS as follows:
  - Define it to a number between 1 and 16.
  - Defining it to 1 turns TELNETSYM into a single-threaded symbiont. Thus, a TELNETSYM crash brings down only the one execution queue in which the crash occurred. Issue:

    ```
    $ DEFINE /SYSTEM UCX$TELNETSYM_STREAMS 1
    ```
  - Define it once. Do not define it differently for each TELNETSYM print queue.
  - The value you enter for UCX$TELNETSYM_STREAMS is passed to the PSM$PRINT system routine in the "streams" argument.
  - If you do not define this logical name, TELNETSYM runs with the OpenVMS maximum of 16 execution queues per symbiont process.

- The following sequence of events now processes correctly:
  1. A job is printed to a TELNETSYM queue, but a link cannot be established to the printer because the printer is not responding (the printer may be busy with another job or out of paper).
  2. You stop this queue with a STOP /QUEUE /RESET command.
  3. A TELNETSYM queue — either another queue or the first one — is started, and the remote printer pointed to by the queue that is started is available.
- If the sequence described in the previous item occurs, and you issue a DELETE /ENTRY command, the queue no longer hangs.

3.15.2 TELNET Client

These problems are fixed:

- Socket options are correctly set against the socket after making a connection.
- The OPEN and CONNECT commands properly accept 5-digit port numbers as one of the parameters.
- The SHOW SESSION command is now implemented.
- TELNET now returns an exit status. If you have multiple sessions, the exit status is from either the last session or the last command you issued.
- The HELP command is fixed.
- The PRINT-SCREEN function is fixed.
- During a SPAWN commmand subprocess, non-privileged users received OPER authorization. This is fixed.
- TN3270 works properly with IBM-3278-4 model terminals.

4
Known Problems

This release of the DEC TCP/IP Services for OpenVMS (UCX) product has the following problems:

- TELNET print symbiont (TELNETSYM)

  If you issue a DELETE /ENTRY command on a job while TELNETSYM is trying to establish a link to a nonresponding the printer, the print queue "hangs." To reset the queue, type:

  ```
  $ STOP /QUEUE /RESET
  ```

- SMTP
  - SMTP does not set the RFC 822 "CC:" header for outbound mail.
  - SMTP does not work correctly when a user, sending mail to multiple users, mixes local and remote addresses. Use the following workaround.

    At the `To:` prompt, use the following syntax:

    ```
    $ MAIL
    MAIL> SEND
    To: SMTP%"addr1,addr2,addr3"
    ```

    where each address is a fully qualified SMTP address.

    Example: Mailing a copy of the same file to:

    - Users `gull` and `swan` on the local system
    - User `tern` on remote host `beach.ocean.com`
    - User `duck` on remote host `lake.tarn.edu`

    ```
    $ MAIL
    MAIL> SEND EGG_HATCHING.TXT
    To: gull,swan,smtp%"tern@beach.ocean.com,duck@lake.tarn.edu"
    Subj:      The Latest in High-Tech Egg Warming
    ```

    To force use of SMTP for the local recipients, keep the local recipients separate from the non-local recipients. At the `To:` prompt, issue:

    ```
    To: SMTP%"gull,swan",smtp%"tern@beach.ocean.com,duck@lake.tarn.edu"
    ```

- NFS Client and Server — OpenVMS-to-OpenVMS mode
  - BACKUP/VERIFY — might issue spurious verification error messages.
  - Remotely accessing CMS libraries via NFS — not recommended.
  - UCX's NFS Client with OpenVMS POSIX — not supported.

    You cannot MAP remote file systems for use by OpenVMS POSIX applications via UCX's NFS Client.

- Installation

The UCX$CLEANUP procedure states that there are no more questions. In actuality, there might two additional questions after this message.

- FTP
  - The ENABLE PORT and DISABLE PORT commands do not work.
  - Angle brackets (< >) are not supported in directory specifications.
  - You cannot use exclamation points (!) in strings or files pointed to by the logical names UCX$FTP_SERVER_ANNOUNCE and UCX$FTP_ANONYMOUS_WELCOME.

5
Restrictions

DEC TCP/IP Services for OpenVMS Version 3.3 has restrictions on the following components:

- NFS Client and Server (Section 5.1)
- NFS Client (Section 5.2)
- UCX on VMScluster systems (Section 5.3)
- BIND Server (Section 5.4)
- Management commands (Section 5.5)

5.1 NFS Client and Server

The NFS Client and Server software have the following restrictions:

- OpenVMS-to-OpenVMS mode

  These restrictions are either specific to OpenVMS-to-OpenVMS mode or are more problematic in this mode:

  – Write sharing

    The NFS server and client running in OpenVMS-to-OpenVMS mode do not prevent opening files for shared write access. However, there is no provision for synchronizing updates among multiple clients or between remote clients and local applications and users. If you have a file simultaneously open for write access by remote users on different OpenVMS client nodes or by remote users and local users, you risk corrupting the data in that file.

  – Identification ACLs

    The rights identifiers shown in ACLs are not correctly converted to or from their ASCII equivalents. They may be displayed in their numeric form, or may appear as the wrong ASCII identifier. For this reason, the server ignores all attempts to write or update ACLs.

  – OpenVMS BACKUP Utility

    You cannot use the /PHYSICAL or /IMAGE qualifiers when backing up from or restoring to a remote NFS-served file system.

    The /VERIFY qualifier might cause inappropriate verification error messages.

  – Running images with kernel mode code

    UCX does not support the running of an image residing on a remote NFS-served disk if that image includes any code that runs in kernel mode. This restriction does not apply to images that call kernel mode routines in shared libraries residing locally on the client node.

- Installing images, page files, or swap files

  It is not supported to specify an image, page file, or swap file that resides on a remote NFS-served disk as the target of any command of the VMSINSTAL or SYSGEN utilities.

- NFS Server
  - The NFS Server keeps access to open files by a remote NFS client. The server closes opens files only if they are not accessed by the client during the INACTIVITY_TIMER interval. Local OpenVMS applications can access the files only after the server closes them.

    Reducing the INACTIVITY_TIMER interval increases the load on the server (more OPEN-CLOSE actions) but also enables local applications to access files with shorter delays.
  - The NFS Server limits access to nonSTREAM_LF files to read-only.

5.2 NFS Client

The NFS Client software differs somewhat from other implementations of this protocol. The main difference is that, on an OpenVMS server, its disk's Master File Directory (MFD) is not world readable.

Both UNIX and OpenVMS NFS Clients can successfully mount the MFD (see Example 5–1).

Example 5–1 Mounting from UNIX and OpenVMS NFS Clients

From the NFS Server:

```
UCX> SHOW MAP


            Dynamic Filesystem Map
  Pathname                 Logical File System
  /dkb100                  DKB100:
```

From the UNIX Client:

```
# mount grackle:/dkb100 /mnt
```

From the OpenVMS Client:

```
UCX> MOUNT DNFS1: /HOST="nfs_server" /PATH="/dkb100"
```

The OpenVMS Client always reads the MFD. However, if it is protected, the operation fails (see Example 5–2).

Example 5–2 Viewing Directories from UNIX and OpenVMS NFS Clients

From the UNIX Client:

```
# ls /mnt
```

... fails (by design) because the MFD is protected.

From the OpenVMS Client:

```
$ DIRECTORY DNFS1:
```

... fails (by design) because the MFD is protected.

From the UNIX Client:

```
# ls /mnt/mydir
```

...executes because the client does not read the MFD.

From the OpenVMS Client:

```
$ DIRECTORY DNFS1:[MYDIR]
```

...also fails (unintentionally) because the OpenVMS client always reads the MFD.

If the MFD of a disk you are mounting is read-protected against the UIC to which the remote OpenVMS clients are mapped ("destination UIC"), do not mount the MFD. Instead, mount the lower-level directories, which are readable by the destination UIC.

5.3 UCX on VMScluster Systems

The UCX configuration procedure, UCX$CONFIG, creates node-specific data on a VMScluster in SYS$COMMON. UCX$CONFIG does not configure this data in SYS$SPECIFIC on a per node basis.

Node-specific data is identified by the member's node name, which is defined in SCSNODE in SYSGEN. If you change your node name in SYSGEN, follow these steps:

1. Delete the node-specific information.
2. Make the changes in SYSGEN.
3. Rerun the UCX$CONFIG procedure.

Avoid binding a socket to a cluster alias.

Because the cluster alias is distributed in time, a socket bound to it has periods during which it is not fully active. A UDP socket may stop to receive data, while a TCP socket may stop both to transmit and receive data.

For receiving data destined to the cluster alias, use the binding to the IP address 0 (UCX$C_INADDR_ANY).

5.4 BIND Server

The BIND Server software has these restrictions:

- You can configure only one primary BIND Server and one secondary server per VMScluster system.
- The BIND Server does not correctly support well known services in the domain (WKS entries) in server databases.

5.5 Management Commands

The management commands have the following restrictions:

- SET NOROUTE command — Partial wildcards are not valid, for example:

  ```
  UCX> SET NOROUTE route*
  ```

- SET SERVICE command — When you modify parameters to a service, disable and re-enable the service for the modifications to take effect.

  Except for TELNET and Remote Login, the DISABLE SERVICE command disables the specified service, but does not stop the current process, if one exists. To stop and restart the current process, follow these steps:

  1. Wait until the process exits or stop it with the DCL STOP PROCESS /ID=*n* command.
  2. Issue ENABLE SERVICE.

6
Documentation

For information about the DEC TCP/IP Services for OpenVMS (UCX) Version 3.3 software, see the following documentation:

- Version 3.3 cover letter
- Version 3.3 *Release Notes*
- Version 3.2 documentation set

  This chapter updates and expands the Version 3.2 manuals.

(Documentation for the new Version 3.3 software comprises Chapter 1.)

6.1 Omitted Documentation

The following information is either missing from, or hard to find in, the Version 3.2 manuals.

6.1.1 TELNET Print Symbiont (TELNETSYM)

The TELNET print symbiont (TELNETSYM) is an OpenVMS-supplied print symbiont. It prints over TCP/IP to a device such as a terminal server that implements TELNET. Its function is similar to that of LATSYM for Digital's Local Area Transport (LAT) software.

TELNETSYM is a true OpenVMS print symbiont that performs all print formatting. (This implementation differs from the LPD Protocol, where the receiver of the print job does the print formatting.)

For information about corrected and enhanced TELNETSYM features, see also Section 3.15.1.

6.1.1.1 Setting Up a Print Queue

Set up a queue to use TELNETSYM just as you would for any print queue, except for these requirements:

1. Specify the TELNETSYM image name in the /PROCESSOR qualifier:

   /PROCESSOR=UCX$TELNETSYM

2. Specify the host and port number to which the queue sends the print data. Use the /ON qualifier:

   /ON="host:port"

   Example: To set up a TELNETSYM queue named `xyz_q` to print to host `printserver.xyz.com` at TCP Port 4242, issue:

   ```
   $ INITIALIZE /QUEUE /PROCESSOR=UCX$TELNETSYM -
   _$ /ON="printserver.xyz.com:4242" xyz_q
   ```

6.1.1.2 Link Establishment

When a print job that has no link to the printer is given to TELNETSYM, the printer attempts to open a link. If a link is established successfully, printing starts. If not, TELNETSYM continues to try to establish a link until it is successful. The configurable period between link-establishment-attempts is the retry interval.

6.1.1.3 Link Shutdown

You can configure TELNETSYM to release an open link in one of two ways:

- Configure the link to release after a period of inactivity, called idle timeout.
- Configure the link to release at the end of a print job.

When multiple systems contend for the same printer, configure TELENETSYM to release the link at the end of a job. This allows other systems to print quickly. The disadvantage is that this creates overhead because it requires a new link for each print job.

When there is little or no contention for a printer, configure TELNETSYM to release the link after an idle timeout has elapsed. With this approach, the TELNETSYM waits for the configured idle time to elapse and then closes the link. This option works well within batch printing applications.

6.1.1.4 Output Byte Stream

The stream of bytes sent to the printer by TELNETSYM differs only slightly from the output of the standard OpenVMS print symbiont. These differences can be controlled with the following TELNETSYM configuration logical names:

- TELNETSYM doubles any TELNET IAC characters found in the byte stream unless UCX$TELNETSYM_RAW_TCP is defined for the queue. The IAC character is a hexadecimal FF.
- TELNETSYM sends the TELNET "do binary will binary" options negotiation escape sequence before a print job that has been printed with /PASSALL, because /PASSALL print jobs may contain 8-bit data. If UCX$TELNETSYM_RAW_TCP is defined for the queue, TELNETSYM does not send this sequence before printing a /PASSALL job. The sequence is six bytes, which are symbolically:

  IAC, DO, BINARY, IAC, WILL, BINARY

  In hexadecimal, this is:

  FF,FD,00,FF,FB,00
- TELNETSYM generates a form feed before the first job of a new print queue starts and at the end of every job. This is normal behavior for an OpenVMS print symbiont. To suppress this behavior, define UCX$TELNETSYM_SUPPRESS_FORMFEEDS for the queue.

6.1.1.5 Configuring Print Queues

You can customize TELNETSYM for each print queue by defining logical names before you start the queue. Because the logical names are translated once at queue startup time, they can be defined differently for each TELNETSYM queue.

Some TELNETSYM configuration logical names are associated with a TELNETSYM configuration option that is either ON or OFF. If the logical name:

- Is defined, the option is ON.

- Is not defined, the option is OFF.

Define all the TELNETSYM configuration logical names as /SYSTEM. The following sections describe the TELNETSYM configuration logical names.

6.1.1.5.1 Stream of Print Bytes Sent Over the Link

- **UCX$TELNETSYM_RAW_TCP** (ON or OFF)

  Suppresses all "TELNET" type modifications of the print output stream. This is for cases where the remote printer/terminal server does not implement the TELNET Protocol and, instead, expects the raw data stream.

  Stops TELNETSYM from doubling IAC characters and sending the TELNET escape sequence to negotiate binary options for files printed /PASSALL.

- **UCX$TELNETSYM_SUPPRESS_FORMFEEDS** (ON or OFF)

  Suppresses form feeds between jobs. This includes the form feed that is normally sent before the very first job printed to a print queue and the form feed sent at the end of every job.

6.1.1.5.2 Diagnostics Logging

- **UCX$TELNETSYM_VERBOSE** (ON or OFF)

  Turns on or off the logging of TELNETSYM diagnostics, for example, messages that indicate when links have come up or gone down, and error messages.

- **UCX$TELNETSYM_NO_OPCOM** (ON or OFF)

  Turns on or off the sending of messages to the operator console.

- **UCX$TELNETSYM_DEBUG**

  Turns on or off TELNETSYM diagnostics to supplement UCX$TELNETSYM_VERBOSE.

  Tells TELNETSYM which diagnostic message types to log.

  Specify a value, which is a bit-mapped longword, where each bit set in the value turns on a particular logging function. The options are:

Bit 0	Tracks the flow of code. Example: such-n-such-routine entered
Bit 1	Tracks the allocation of memory. Example: about to call malloc Example: just freed address 7F0000
Bit 2	Logs the bytes sent and received over TCP/IP link.

  To set a bit, assign the value to the logical name whose binary equivalent would have the bit set. For example, you can tell TELNETSYM to log everything that it writes to, and receives from, the TCP/IP link by issuing:

```
$ DEFINE /SYSTEM UCX$TELNETSYM_DEBUG 4
```

  Decimal 4 is binary 100 with bit 2 set. Note that different combinations can be achieved by setting more than one bit in the value. A value of 3, for example, sets bits 0 and 1 and so causes logging of flow of code and memory allocation diagnostics.

  If UCX$TELNETSYM_DEBUG is undefined, TELNETSYM does not log these diagnostics.

Bit 2 is useful in unassisted problem solving. Bits 0 and 1 are primarily for Digital use. However, with knowledge of PSM$ symbionts, you might find all the options useful.

- **UCX$TELNETSYM_LOG_KEEP**

  Specify a value to set the number of copies of a log file that TELNETSYM stores before it starts purging them.

  If you do not define UCX$TELNETSYM_LOG_KEEP, TELNETSYM does not purge.

- **UCX$TELNETSYM_SCRATCH**

  Defines the directory for TELNETSYM to use as its scratch directory.

  TELNETSYM puts in this directory log files and temporary files created by TELNETSYM relay queues.

  - If UCX$TELNETSYM_SCRATCH is not defined, the UCX$LPD_SPOOL directory is used.
  - If UCX$LPD_SPOOL is not defined, SYS$SPECIFIC:[SYSEXE] is used.

6.1.1.5.3 Characteristics of the TCP/IP Link

The TELNETSYM configuration logical names in this category correspond directly to item list options for the UCX $QIO `setmode` function (for details, see the *DEC TCP/IP Services for OpenVMS System Service and C Socket Programming* manual).

- **UCX$TELNETSYM_DROPTIME**

  TELNETSYM sets the UCX$C_TCP_DROP_IDLE $QIO `setmode` item to the value of UCX$TELNETSYM_DROPTIME if it is defined.

- **UCX$TELNETSYM_KEEPALIVE** (ON or OFF)

  If you define UCX$TELNETSYM_KEEPALIVE, TELNETSYM sets the $QIO `setmode` UCX$C_KEEPALIVE option.

- **UCX$TELNETSYM_PROBETIME**

  TELNETSYM sets the UCX$C_TCP_PROBE_IDLE $QIO `setmode` item to the value of UCX$TELNETSYM_PROBETIME, if it is defined.

- **UCX$TELNETSYM_SNDBUF**

  TELNETSYM sets the UCX$C_SO_SNDBUF $QIO `setmode` item to the value of UCX$TELNETSYM_SNDBUF, if it is defined.

Note that the value for drop time must be greater than the value for probe time. When you define only one of these TELNETSYM logicals, the value of the other comes from the UCX default value. To see this default, issue SHOW PROTOCOL TCP /PARAMETER.

6.1.1.5.4 Establishment and Release of the TCP/IP Link

- **UCX$TELNETSYM_RETRY_INTERVAL**

  Defines the time for TELNETSYM to wait between link-establishment retries when link establishment has failed. The value for this logical name is an OpenVMS delta time.

  If this logical name is not defined, TELNETSYM defaults to a wait period of three minutes between retries.

Example: To define a retry interval of 30 seconds, issue:

```
$ DEFINE /SYSTEM UCX UCX$TELNETSYM_RETRY_INTERVAL "0 00:00:30.00"
```

- **UCX$TELNETSYM_IDLE_TIMEOUT**

  Defines the length of time after which TELNETSYM terminates an inactive link. Specify a value that is an OpenVMS delta time. If you:

  - Define UCX$TELNETSYM_IDLE_TIMEOUT, TELNETSYM disconnects the link only when a period of inactivity on the link equal to the time specified has passed.
  - Do not define UCX$TELNETSYM_IDLE_TIMEOUT, TELNETSYM releases the link immediately after the completion of each print job.

  Example: To define a link-idle-timeout of ten minutes, issue:

```
$ DEFINE /SYSTEM UCX UCX$TELNETSYM_IDLE_TIMEOUT "0 00:10:00.00"
```

6.1.1.5.5 Other

UCX$TELNETSYM_STREAMS

Configures the number of execution queues handled by each TELNETSYM process. Define this logical name as follows:

- Define it to a number between 1 and 16.
- Defining it to 1 turns TELNETSYM into a single-threaded symbiont.
- Define it once. Do not define it differently for each TELNETSYM print queue.

6.1.1.6 Relay Queues

You can redirect the output of TELNETSYM to another queue, rather than directly sending it to a remote printer. A queue with this setup is a "relay queue." Use relay queues to funnel fully formatted output to an outbound LPD queue. LPD transfers jobs that are fully formatted on the sending side by OpenVMS.

TELNETSYM saves the output stream to a temporary file and then submits the file to the destination queue. TCP/IP is not used.

To set up a TELNETSYM relay queue, specify the /ON qualifier of the INITIALIZE/QUEUE command as:

`/ON="UCX$QUEUE:`*`qname`*`"`

where:

qname is the name of the queue that you want the TELNETSYM to send its output to.

Example: To set up a TELNETSYM relay queue named RELAYQ_4 to send output to the queue named LPD_Q4, issue:

```
$ INITIALIZE /QUEUE /ON="UCX$QUEUE:LPD_Q4" -
_$ /PROCESS=UCX$TELNETSYM /DEVICE=PRINTER RELAYQ_4
```

6.1.1.7 Common Problems

Ensure that you have addressed the following issues:

- **The first use of UCX$TELNETSYM on systems formerly using the public domain TELNET symbiont**

  If you use the public domain TELNET symbiont and want to switch to the UCX TELNET symbiont, remember to change the value of /PROCESSOR on the TELNET symbiont queues, including any command procedures that start up the queues. Change:

  ```
  /PROCESSOR=TELNETSYM
  ```

  to:

  ```
  /PROCESSOR=UCX$TELNETSYM
  ```

- **Printing to Digital terminal servers**

  When you print to a DECserver system, ensure that:

  – Input flow control for the port you are printing to is set to DISABLED. Issue:

    ```
    $ CHANGE PORT port INPUT FLOW DISABLED
    ```

  – The TELNET server for the terminal server port is properly set. Issue:

    ```
    $ CHANGE PORT port -
    _$ TELNET SERVER NEWLINE TO HOST <CRLF>
    ```

- **Print queue in "stalled" state**

  When you print a job to a TELNETSYM queue, a link must be established between the queue and the printer. If there is high contention for the printer, it might be busy, causing the first attempt at link-establishment to fail.

  TELNETSYM continues to try to establish the link, according to the retry interval logical. Until the link is established, the execution queue stalls. When the link comes up, the job prints. A "stalled" TELNETSYM queue is not necessarily an error.

  If the queue stalls while printing a job, the printer is probably out of paper.

- **TELNETSYM logging**

  Each TELNETSYM queue has a log file named UCX$TELNETSYM_ *qname*.LOG. If you:

  – Define the logical name UCX$TELNETSYM_SCRATCH, the log files are stored in the UCX$TELNETSYM_SCRATCH directory.

  – Do not define UCX$TELNETSYM_SCRATCH, the log files go into UCX$LPD_SPOOL.

    If UCX$LPD_S POOL is not defined, the logs go into SYS$SPECIFIC:[SYSEXE].

- **Format problems**

  To track down problems with wrong formatting on the printed page (for example, "garbage" for a graphics file or unwanted blank pages), use Bit 2 of the the TELNETSYM logical name UCX$TELNETSYM_DEBUG. Defining UCX$TELNETSYM_DEBUG helps determine whether or not the source of the problem is TELNETSYM. Follow these steps:

  1. Define UCX$TELNETSYM_DEBUG to 4 in the system table. Issue:

```
$ DEFINE /SYSTEM UCX$TELNETSYM_DEBUG 4
$ STOP /QUEUE /RESET TELNETSYM_qname
$ START /QUEUE TELNETSYM_qname
```

2. Print the job that does not print properly.
3. Look at the TELNETSYM log file for the queue.

   This file has messages that show you every byte sent over the link to the printer, such as control characters and setup/reset modules.

   If the raw TCP logical name is not defined, you see doubled IAC characters (hexadecimal FF).

   If you printed /PASSALL and the raw TCP logical is not defined, the job starts with the TELNET options negotiation sequence "do binary, will binary".
4. Identify the problem. Either fix it or report it to Digital.
5. Start the TELNETSYM queue.

- **TELNETSYM buffer dumps**

  TELNETSYM logs control characters and non-printing characters by preceding the hexadecimal value of the byte with a backslash. For example, this sequence:

  ```
  Carriage Control
  Form Feed
  Carriage Control
  Line Feed
  Tab
  the text "Use Your Screen Saver to Conserve Energy."
  Carriage Return
  Line Feed
  ```

  is logged as:

  ```
  \0D\0C\0D\0A\09Use Your Screen Saver to Conserve Energy.\0D\0A
  ```

  The "do binary will binary" sequence starting off a /PASSALL job appears as:

  \FF\FD\00\FF\FB\00

6.1.2 FTP Logical Names

To manage FTP, the software includes the following logical names:

- UCX$FTPD_SECURITY

  If you define UCX$FTPD_SECURITY systemwide, FTP prompts for a password, even if there is not a user name to match the one specified by the FTP user.

- UCX$FTP_NO_VERSION

  If you define UCX$FTP_NO_VERSION, FTP does not send file version numbers when you issue the `mget` and the `ls` commands to a non-OpenVMS host.

- UCX$FTP_ANONYMOUS_WELCOME and UCX$FTP_SERVER_ANNOUNCE

  You can define the welcome message and announcement message by defining the logical name to be equivalent to a string or by storing the text in a file.

  – Using an equivalence string:

Enclose the message text in parentheses. Use this syntax:

```
$ DEFINE /SYSTEM /EXECUTIVE UCX$FTP_ANONYMOUS_WELCOME -
_$ "text"
```

– In a file:

Create the text file.

When you define the logical name, specify this file name. Use this syntax:

```
$ DEFINE /SYSTEM /EXECUTIVE UCX$FTP_SERVER_ANNOUNCE -
_$ "@SYS$SYSDEVICE:[UCX$FTP]file_name"
```

6.1.3 BIND SERVER Error Message

The error message:

```
No root nameservers for class 1
```

means that no root name servers were found because no cache data was available. Either the cache file is not installed on your host, or it has been deleted. Without these records the UCX name server will be unable to resolve names outside of its authoritative data.

Check SYS$SPECIFIC:[UCX$BIND]NAMED.CA. Look for the record using:

UCX SHOW CONFIG BIND

To add this record. Issue:

```
UCX> SET CONFIGURATION BIND /CA
```

Example 6–1 shows a typical snapshot of a secondary BIND Server.

Example 6–1 Typical SHOW CONFIGURATION BIND Display

```
Primary
  Domain:     0.0.127.IN-ADDR.ARPA          File: NAMED.LOCAL

Secondary
  Domain:     UCX.LKG.DEC.COM               File: UCX_LKG_DEC_COM.DB
  Host:       16.20.208.53

Secondary
  Domain:     208.20.16.IN-ADDR.ARPA        File: 208_20_16_IN-ADDR_ARPA.DB
  Host:       16.20.208.53

Cache
  Domain:     .                             File: NAMED.CA
```

6.1.4 Configuring SMTP

Part of configuring SMTP is using the SET MX_RECORDS command. Before you issue this command, ensure that you understand how SMTP routes mail. Detailed examples are in the description of the SHOW MX_RECORDS command.

6.2 Known Errors

The UCX Version 3.2 documentation set has the following problems.

6.2.1 "DEC TCP/IP Services for OpenVMS Management" Manual

6.2.1.1 Table 6-2

- **Incorrect:** The information about UCX$FTP_WINDSIZ says to see Section 6.2 for information about setting a default buffer size.
- **Correct:** See Section 6.5.1.

6.2.1.2 Section 6.6.4

- **Incorrect:**

  Note

  The public directories are read-only, so GET commands work there and PUT commands do not.

  There are limits on the number of files in the user's home directory. Also, the maximum transfer size is restricted.
- **Correct:** Delete this information. Both directory and file access are controlled by OpenVMS access methods.

6.2.1.3 Section 9.1.1

Organization of the Proxy Database — second bullet

- **Incorrect:** The system manager can define the Type field of a Proxy Database entry with up to 3 letters: O (outgoing), N (incoming), and D (dynamic).
- **Correct:** The system manager can define the Type field of a Proxy Database entry with up to 2 letters: O (outgoing) and N (incoming).

6.2.1.4 Section 9.3

Adhere to the Rules for Accessing an NFS Client:

- The third bullet states that if there is a one-to-one correspondence between the OpenVMS account and the UID/GID pair, the process chooses this entry.
- Incorrectly omitted was the statement that the host names must match for the process to choose the entry.

6.2.1.5 Table 9-2

Some items in the table refer to the previous page.

- **Incorrect:** The second entry in the table instructs the system manager to map requests from UID 115 to OpenVMS account USER2.
- **Correct:** The system manager should map requests from UID 115 to OpenVMS account USER3.

6.2.1.6 Section 9.5.7

MOUNT command, Examples, Example 2:

- **Incorrect:**

  ```
  UCX> SHOW DEVICE DNFS3: /FULL
  ```

- **Correct:**

  ```
  UCX> SHOW MOUNT DNFS3: /FULL
  ```

- **Incorrect:** Directory name DNFC3.
- **Correct:** Directory name DNFS3.

6.2.1.7 Section 9.5.8.2

Creating a customized Attributes Data Files, Step 4:

- **Incorrect:**

  ```
  $ COPY GAF.ADF SYS$SYSTEM
  ```

- **Correct:**

  ```
  $ COPY GAF.ADF SYS$SYSTEM:UCX$GAF.ADF
  ```

6.2.2 "DEC TCP/IP Services for OpenVMS Management Command Reference" Manual

The *DEC TCP/IP Services for OpenVMS Management Command Reference* manual contains the following errors.

6.2.2.1 NFS Server: MAP-Related Command Descriptions

MAP, NOMAP, SET CONFIGURATION MAP, SET CONFIGURATION NOMAP, and GENERATE MAP command descriptions:

- **Incorrect:** The mapping (formerly BIND) command descriptions fail to make clear the information for temporarily mapping file systems and entering mapping information into the permanent Configuration Database.
- **Correct:**
  - MAP

    Created logical file systems are not permanent.
  - NOMAP

    Temporarily unlinks existing logical file systems.
  - SET CONFIGURATION MAP, GENERATE MAP

    Created logical file systems are entered into the permanent Configuration Database.
  - SET CONFIGURATION NOMAP

    Permanently unlinks existing logical file systems in the Configuration Database.

6.2.2.2 CONVERT /ULTRIX HOST Command

Description of the *dest_file* parameter:

- **Incorrect:** The default file name is SYS$COMMON:[SYSEXE]ETC.HOSTS.
- **Correct:** The default file name is ETC.HOSTS in the directory with the BIND Server.

6.2.2.3 SET BOOTP Command

Description of the /SERVERS=*type*=*host* qualifier:

- **Incorrect:** [NO]RESERVED
- **Correct:** Omit this value.

6.2.2.4 SET MX_RECORDS Command

The description of the /PREFERENCE=*n* qualifier incorrectly states that /PREFERENCE is optional. The /PREFERENCE=*n* qualifier is required.

6.2.2.5 SET CONFIGURATION NAME_SERVICE Command

/LOCAL qualifier — included by mistake. This qualifier is not supported.

6.2.2.6 SET CONFIGURATION NONAME_SERVICE Command

Mistakenly omitted.

6.2.2.7 SET SERVICE Command

Command description — Note

- **Incorrect:** SET NOSERVICE with neither a specified service nor specified qualifiers deletes all entries for all services.
- **Correct:** Delete this statement.

Description of the *service* parameter:

- **Incorrect:** The system manager can specify up to 15 characters.
- **Correct:** The system manager can specify up to 10 characters.

6.2.2.8 MOUNT Command

First bullet in the Notes in the command description:

- **Incorrect:** If you MOUNT remote OpenVMS directories where the NFS Server is UCX, use the /NOADF qualifier on the MOUNT command line.
- **Correct:** If you MOUNT remote OpenVMS directories where the NFS Server is UCX, use the /NOADF qualifier on the MOUNT command line unless you are using the VMS-to-VMS Integration feature.

  The requirement for the /NOADF qualifier applies only if the remote UCX server is a version older than Version 3.3 and cannot participate in OpenVMS-to-OpenVMS mode operation.

7 Unsupported Components

The following software, included with the DEC TCP/IP Services for OpenVMS (UCX) Version 3.3 release, is not supported by Digital:

- Outbound TELNET devices (Section 7.1)
- Remote File Copy (RCP) (Section 7.2)
- Incoming RLOGIN proxies (Section 7.3)

Warning

For these unsupported components, Digital makes no representations regarding their use or performance. Digital accepts no responsibility for any expenses, losses, or action incurred or undertaken by any party as a result of using of this software. In no event shall Digital have any liability whatsoever for incidental and consequential damages as a result of the performance, use, or operation of this software.

7.1 Support for Outbound TELNET Devices

You can create a TELNET pseudo-device (virtual terminal) and connect it to a remote listener (port). Use the following TELNET command:

```
TELNET> CREATE_SESSION host port [unit]
```

where:

host Specifies the remote host name.

port Specifies the listening port number at the remote end.

unit Optional. A decimal number specifying the unit number for the pseudo terminal (TN*x*).

The unit "0" is the default and has a special meaning allowing the system to pick the next available unit. If the requested unit is already in use, the next available unit is chosen, and the command completes successfully. In any case, the user is notified of the unit number created.

```
TELNET> DELETE_SESSION unit
```

where:

unit A decimal number specifying the the unit number to be deleted.

If the TN specified has been set via CREATE_SESSION, it is disconnected and deleted. Otherwise, an error is returned.

7.2 Remote File Copy (RCP)

Remote Copy (RCP command) copies files between internet hosts.

RCP

Copies files between internet hosts. Issue the RCP command at the DCL prompt. You can copy files:

- From a remote host to your host
- From your host to a remote host
- From one remote host to another

To use RCP, the Remote Shell Service must be enabled.

You have two ways to set up proxy checking:

- On the RCP command line, specify a remote user name and, optionally, a password.
- Add proxies to the Proxy Database.

  For remote UNIX hosts, add an entry in the host equivalence file, `/etc/hosts.equiv`.

You can specify qualifiers in either OpenVMS-Style format or UNIX-Style format, but do not mix both types on the same command line.

DCL-Style Format

RCP *source_file destination_file*

UNIX-Style Format

rcp */source.file /destination.file*

Parameters

source_file
Required.

Source host and file specification, in the format *`[user@]host:file`*, where:

- *user@* is the user name on the remote UNIX system.

  Needed only if the UNIX system has the name in its `/etc/hosts.equiv` file or the UNIX user's `.rhosts` file.
- *host* is the remote host, followed by a colon (:).
- *file* is the name of the file to copy. A file name without the full path specification defaults to the default (or home) directory. Table 7–1 shows the possible correct formats.

Table 7–1 RCP Command: Specifying the Source File

Host	Possible Formats
UNIX hosts	• Absolute path name, such as `/etc/user/hosts`, followed by the file name • Path name relative to your default directory, followed by the file name
OpenVMS hosts	• Brackets ([ ]) (indicates your default directory), followed by the file name • Full file specification, such as DKA0:[WILDLIFE.BIRDS.NORTHERN]BIG_BIRDS.TXT To specify a device name, type a colon (:) and then the name. Enclose the entire parameter within quotation marks (" "). • A logical name, such as SYS$LOGIN:ROBIN.DAT, BIRDS$9:[AMERICAN]FINCHES.LIS To specify a logical name, type a colon (:) and then the name. Enclose the entire parameter within quotation marks (" ").

destination
Required.

Destination host and file specification information is of the same form as the source parameter, except:

> If the file specification is completely omitted or the file name portion of the file specification is omitted, the default file name used is the same as specified in the source parameter, the directory being the default/home directory of the user.

Qualifiers

/LOG
Optional. Default: no logging.

Logs the files copied to or from the local system.

/PRESERVE (OpenVMS-Style)
-p (UNIX-Style)
Optional.

Preserves the file protection mode and modification date during a copy.

/RECURSIVE (OpenVMS-Style)
-r (UNIX-Style)
Recursively copies each subtree rooted at the directory you specify in the UNIX file specification. For OpenVMS hosts, specify [dir...] (with three trailing ellipses) in the filespec instead of using this qualifier.

/USER=*username*

User on the remote host. Use only if a proxy has not been added for this user on the remote host, or if the remote host's host equivalence file does not contain an entry for this user. If necessary, truncate the user name to the required number of characters using the /TRUNCATE qualifier below. Specifying "username@" with the source or destination parameter is the equivalent UNIX-style method.

/PASSWORD=*password*

Password on the destination system. Use with the /USER qualifier.

/TRUNCATE[=*n*]

Optional. Default: no truncation.

Truncates the user name to the specified number of characters. If you omit *n*, the default is eight characters.

Examples

1. `$ RCP /LOG unixhost:source_file []`

   Copies a file from a remote UNIX system under its home directory to a local file of the same name in the current directory. The log information for the copy is displayed.

2. `$ RCP vmshost:source_file [dir]dest_file`

   Copies a file from a remote VMS system under its home directory to a local file of a different name in the specified directory.

3. `$ RCP /PRESERVE vmshost:[dir1]source_file ":dka300:[dir2]"`

   Copies a file from a remote VMS system from the directory specified to the directory on the device system on the local system into the same file name. The copy preserves the source file's protection mode and modification date.

   Use quotation marks (" ") for specifying the device and directory on the destination.

4. `$ RCP /USER=user1 /PASS=pass1 ":SYS$LOGIN:source_file" unix_host:`

   Copies a file from your login directory of the local system to a remote UNIX system into the user's login directory. The user name and password on the UNIX system are also specified.

5. `$ rcp -r ":dka300:[source_dir...]" "unixhost:/usr/tmp/"`

   Copies all files under the local directory specified to a remote UNIX host's destination directory. If there are subdirectories beneath the source directory specified, all the files in them are copied as well, creating subdirectories on the remote host, as appropriate. Directory hierarchy on a UNIX host is preserved by default.

6. `$ RCP /LOG /RECURSIVE [source_dir...] vmshost:[dest_dir...]`

   Copies the complete local sub-directory tree specified, to a remote VMS host's destination directory while preserving the directory hierarchy and logging each file copy.

7. `$ RCP /LOG /RECURSIVE [source_dir...] vmshost:[dest_dir]`

   Same as Example 6 except that the copy is done to the destination directory itself without preserving directory hierarchy (new sub-directories are not created).

8. ```
   $ RCP /USER=someone /PASSWORD=password -
   _$ /TRUNCATE=4 source_file host:dest_file
   ```

   Copies the local file to a remote user's login directory. Note the truncation of the remote user name. A user name and password are necessary if proxy records or host equivalence file entries are not present.

9. `$ RCP host1:file1 host2:file2`

   Copies a file from one remote host to another. Assumes proxy records or host equivalence files are set up correctly for user executing this command.

10. `$ RCP "user1@host1:sys$dir:file1" "user2@host2:/usr/user2"`

    Copies a file from remote OpenVMS system `host1` in the directory pointed to by the logical name SYS$DIR to remote UNIX system host2 in the specified directory. Different user names are used on the two remote systems. Proxy records and host equivalence files must be set up properly because the passwords are not being passed.

11. `$ RCP /USER=username /PASSWORD=password host1:[dir]source_file "host2:/usr"`

    Copies a file from an OpenVMS host to a UNIX host, following the directory specifications. The same user name and password are used on both remote systems.
   ```

7.3 Incoming RLOGIN Proxies

UCX now supports incoming RLOGIN proxies. If you set the proxy flag for the RLOGIN Service, inbound requesters with valid proxies are automatically logged in, thus eliminating "Username:" and "Password:" prompting. (Users without a valid proxy are still prompted.)

To set the proxy flag, issue:

```
UCX> SET SERVICE RLOGIN /FLAGS=(PROXY)
```

To add a proxy for an incoming RLOGIN user to the Proxy Database, issue:

```
UCX> ADD PROXY user /HOST=host /REMOTE_USER=user [ /PERMANENT ]
```

Example: This command adds to the Proxy Database the proxy REMOTE_UGLY_DUCKLING for incoming RLOGIN user `cygnet` at host `swan`.

```
UCX> ADD PROXY REMOTE_UGLY_DUCKLING -
_UCX> /REMOTE_USER="cygnet" /HOST="babyswan"
```

A
Installed Files

The installation procedure installs the following software.

A.1 UCX Files

```
SYS$COMMON:[SYSEXE]UCX$SNMP_AGENT.EXE
SYS$COMMON:[SYSEXE]UCX$VERSIONS.EXE
SYS$COMMON:[SYSEXE]UCX$UCP.EXE
SYS$COMMON:[SYSEXE]UCX$PING.EXE
SYS$COMMON:[SYSLIB]UCX$IPC_SHR.EXE
SYS$COMMON:[SYSTEST.UCX]UCX$INET_IVP.EXE
SYS$COMMON:[SYSMSG]UCX$MSG.EXE
SYS$COMMON:[SYS$LDR]UCX$BGDRIVER.EXE
SYS$COMMON:[SYS$LDR]UCX$INTERNET_SERVICES.EXE
SYS$COMMON:[SYS$LDR]UCX$INTERNET_SERVICES_V6.EXE
SYS$COMMON:[SYS$LDR]UCX$INETDRIVER.EXE
SYS$COMMON:[SYSLIB]UCX$ACCESS_SHR.EXE
SYS$COMMON:[SYSEXE]UCX$INETACP.EXE
SYS$COMMON:[SYSEXE]UCX$INET_ROUTING.EXE
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$INTERNET_SERVICES_SEC.EXE
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$INTERNET_SERVICES_V6_SEC.EXE

(moved from UCX$EXAMPLES:) - SYS$COMMON:[SYSEXE]UCX$TRACE.EXE
(new) - SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TRACE_HAS_MOVED.EXE
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]TRACEROUTE.EXE

SYS$COMMON:[SYSHLP.EXAMPLES.UCX]BUILD_UCX_SECURITY_DRIVER.COM
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX_SECURITY_DRIVER.MAR
SYS$COMMON:[SYSMGR]UCX$BOOTP_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$BOOTP_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$CALLBACKS.COM
SYS$COMMON:[SYSUPD]UCX$CLEANUP.COM
SYS$COMMON:[SYSMGR]UCX$CONFIG.COM
SYS$COMMON:[SYSMGR]UCX$DNFS_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$FIXUP.COM
SYS$COMMON:[SYSEXE]UCX$FTPSERVER.COM
SYS$COMMON:[SYSMGR]UCX$FTPD_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$FTPD_STARTUP.COM
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$IOCTL_ROUTINE.C
SYS$COMMON:[SYSLIB]UCX$IPC.OLB
SYS$COMMON:[SYSLIB]UCX$INETDEF.H
SYS$COMMON:[SYSLIB]UCX$INETDEF.R32
SYS$COMMON:[SYSLIB]UCX$INETDEF.MAR
SYS$COMMON:[SYSLIB]UCX$INETDEF.PAS
SYS$COMMON:[SYSLIB]UCX$INETDEF.FOR
SYS$COMMON:[SYSLIB]UCX$INETDEF.PLI
SYS$COMMON:[SYSLIB]UCX$INETDEF.ADA
SYS$COMMON:[SYSLIB]UCX$INETDEF.BAS
SYS$COMMON:[SYSMGR]UCX$INET_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$INET_STARTUP.COM
SYS$COMMON:[SYSTEST]UCX$IVP.COM
SYS$COMMON:[SYSMGR]UCX$LOAD_INETDRIVER.COM
SYS$COMMON:[SYSMGR]UCX$LPD_RCV_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$NFS_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$NFS_SERVER_STARTUP.COM
```

```
SYS$COMMON:[SYSMGR]UCX$PORTM_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$PORTM_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$PRINTCAP.DAT
SYS$COMMON:[SYSMGR]UCX$REMOTE_TTY_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$REXECD_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$RSHD_STARTUP.COM

(new) - SYS$COMMON:[SYSMGR]UCX$NTPD_STARTUP.COM
(new) - SYS$COMMON:[SYSMGR]UCX$NTPD_SHUTDOWN.COM

SYS$COMMON:[SYSMGR]UCX$SMTP_RECV_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$SMTP_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$SMTP_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$SNMP_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$SNMP_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$SNMPD_MIR.DAT
SYS$COMMON:[SYSMGR]UCX$SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$SERVICE_SETUP.COM
SYS$COMMON:[SYSMGR]UCX$SYMBOLS.COM
SYS$COMMON:[SYSMGR]UCX$REGISTER.COM
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_CLIENT_IPC.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_CLIENT_QIO.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_CLIENT_QIO.MAR
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_SERVER_IPC.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_SERVER_IPC_AUXS.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_SERVER_QIO.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_SERVER_QIO_AUXS.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$TCP_SERVER_QIO.MAR
SYS$COMMON:[SYSMGR]UCX$TELNET_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$TFTP_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$TFTP_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$TIMEZONE.COM
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]TN3270DEF.MAR
SYS$COMMON:[SYSHLP]UCX$UCP_HELP.HLB
SYS$COMMON:[SYSMGR]UCX$UCP_STARTUP.COM
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$UDP_CLIENT_IPC.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$UDP_CLIENT_QIO.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$UDP_CLIENT_QIO.MAR
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$UDP_SERVER_IPC.C
SYS$COMMON:[SYSHLP.EXAMPLES.UCX]UCX$UDP_SERVER_QIO.C
SYS$COMMON:[SYSMGR]UCX$BIND_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$BIND_SHUTDOWN.COM
SYS$COMMON:[SYSMGR]UCX$PCNFSD_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$PCNFSD_SHUTDOWN.COM
SYS$COMMON:[SYSEXE]UCX$INTERNET_SERVICES.STB
SYS$COMMON:[SYSEXE]UCX$INTERNET_SERVICES_V6.STB
SYS$COMMON:[SYSEXE]UCX$INTERNET_SERVICES_SEC.STB
SYS$COMMON:[SYSEXE]UCX$INTERNET_SERVICES_V6_SEC.STB
SYS$COMMON:[SYSEXE]UCX$INETACP.STB
SYS$COMMON:[SYSEXE]UCX$INET_ROUTING.STB
SYS$COMMON:[SYSEXE]UCX$NET_GLOBALS.STB
```

```
SYS$COMMON:[SYSLIB]UCX$CFS_SHR.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSMOUNT_V5.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSMOUNT_V6.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSDISMOUNT_V5.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSDISMOUNT_V6.EXE
SYS$COMMON:[SYS$LDR]UCX$DNFSDRIVER_V5.EXE
SYS$COMMON:[SYS$LDR]UCX$DNFSDRIVER_V6.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSACP_V5.EXE
SYS$COMMON:[SYSEXE]UCX$DNFSACP_V6.EXE
SYS$COMMON:[SYSEXE]UCX$PCNFSD.EXE
SYS$COMMON:[SYSEXE]UCX$PORTMAPPER.EXE
SYS$COMMON:[SYSLIB]UCX$RPCXDR_SHR.EXE
SYS$COMMON:[SYSEXE]UCX$RPCINFO.EXE
SYS$COMMON:[SYSTEST.UCX]UCX$RPCIVP_CLIENT.EXE
SYS$COMMON:[SYSTEST.UCX]UCX$RPCIVP_SERVER.EXE
SYS$COMMON:[SYSEXE]UCX$SERVER_NFS.EXE
SYS$COMMON:[SYSEXE]UCX$CONVERT.FDL
SYS$COMMON:[SYSEXE]UCX$CONVERT.COM
SYS$COMMON:[SYSHLP]UCX$VMS_FILES.DOC
SYS$COMMON:[SYSLIB]UCX$RPCXDR.H
SYS$COMMON:[SYSEXE]UCX$EXE.ADF
SYS$COMMON:[SYSEXE]UCX$HLB.ADF
SYS$COMMON:[SYSEXE]UCX$MLB.ADF
SYS$COMMON:[SYSEXE]UCX$OBJ.ADF
SYS$COMMON:[SYSEXE]UCX$OLB.ADF
SYS$COMMON:[SYSEXE]UCX$STB.ADF
SYS$COMMON:[SYSEXE]UCX$TLB.ADF
SYS$COMMON:[SYSEXE]UCX$BOOTP.EXE
SYS$COMMON:[SYSEXE]UCX$FTP.EXE
SYS$COMMON:[SYSEXE]UCX$FTPC.EXE
SYS$COMMON:[SYSEXE]UCX$FTPD.EXE
SYS$COMMON:[SYSEXE]UCX$LPD_RCV.EXE
SYS$COMMON:[SYSLIB]UCX$LPD_SHR.EXE
SYS$COMMON:[SYSEXE]UCX$LPD_SMB.EXE
SYS$COMMON:[SYSEXE]UCX$LPRM.EXE
SYS$COMMON:[SYSEXE]UCX$LPRSETUP.EXE
SYS$COMMON:[SYSEXE]UCX$LPQ.EXE
SYS$COMMON:[SYSEXE]UCX$TELNETSYM.EXE
SYS$COMMON:[SYSEXE]UCX$NSLOOKUP.EXE
SYS$COMMON:[SYSEXE]UCX$RLOGIN.EXE
SYS$COMMON:[SYSEXE]UCX$RSH.EXE

(new) - SYS$COMMON:[SYSEXE]UCX$RCP.EXE
(new) - SYS$COMMON:[SYSEXE]UCX$NTPD.EXE

SYS$COMMON:[SYSLIB]UCX$SMTP_PARSESHR.EXE
SYS$COMMON:[SYSLIB]UCX$SMTP_MAILSHR.EXE
SYS$COMMON:[SYSEXE]UCX$SMTP_RECEIVER.EXE
SYS$COMMON:[SYSEXE]UCX$SMTP_SYMBIONT.EXE
SYS$COMMON:[SYSEXE]UCX$TELNET.EXE
SYS$COMMON:[SYS$LDR]UCX$TNDRIVER.EXE
SYS$COMMON:[SYSEXE]UCX$TFTP.EXE
SYS$COMMON:[SYSEXE]UCX$UUENCODE.EXE
SYS$COMMON:[SYSEXE]UCX$UUDECODE.EXE
SYS$COMMON:[SYSEXE]UCX$BIND_SERVER.EXE
SYS$COMMON:[SYSEXE]UCX$BIND_SERVER_XFER.EXE
SYS$COMMON:[SYSHLP]UCX$FTP_HELP.HLB
SYS$COMMON:[SYSHLP]UCX$TELNET_HELP.HLB
SYS$COMMON:[SYSHLP]UCX$NSLOOKUP_HELP.HLB
SYS$COMMON:[SYSEXE]UCX$ENCODE.COM
SYS$COMMON:[SYSEXE]UCX$DECODE.COM

(new) - SYS$COMMON:[SYSEXE]UCX$TNDRIVER.STB
```

A.2 Files Installed for PATHWORKS Systems

```
SYS$COMMON:[SYS$LDR]UCX$PWIPDRIVER.EXE
SYS$COMMON:[SYSEXE]UCX$PWIPACP.EXE
SYS$COMMON:[SYSEXE]UCX$PWIPSHUT.EXE
SYS$COMMON:[SYSMGR]UCX$PWIP_STARTUP.COM
SYS$COMMON:[SYSMGR]UCX$PWIP_SHUTDOWN.COM
SYS$COMMON:[SYSEXE]UCX$PWIPDRIVER.STB
SYS$COMMON:[SYSEXE]UCX$PWIPSYM.STB

(new) - SYS$COMMON:[SYSLIB]UCX$SDA.EXE
```