SCO® TCP/IP Runtime System for SCO® UNIX® Systems

User's and Administrator's Guide

SCO®

OPEN SYSTEMS SOFTWARE

SCO® TCP/IP Runtime System for SCO UNIX® Systems

Release and Installation Notes

SCO TCP/IP is derived from Lachman Technology, Inc. (LTI)
SYSTEM V STREAMS TCP/IP, a joint development of LTI and Convergent Technologies.
SCO NFS was developed by Lachman Technology, Inc.
SCO NFS is derived from LTI SYSTEM V NFS, a joint development of LTI and Sun Microsystems.

Date: 27 August 1993
Document Version: 1.2.1D

Introduction 1

Chapter 1

Contents and features of this package 3

Chapter 2

System requirements 9

Chapter 3

Installing and configuring SCO TCP/IP 11

Introduction

SCO® TCP/IP for UNIX System V/386 Release 1.2.1 is an implementation of SCO TCP/IP and related protocols for SCO UNIX® Release 3.2 Version 4.2. The product, although based on the latest functional and performance improvements of 4.3BSD UNIX, has been adapted to run within the STREAMS framework and TLI specification of SCO UNIX.

This release of SCO TCP/IP is intended for SCO UNIX Release 3.2 Version 4.2 systems only and not for previous releases of the SCO UNIX system or SCO XENIX System V.

> **NOTE** Please read through this document before installing the SCO TCP/IP software.

Organization of these notes

These notes are organized as follows:

- Chapter 1 describes product features
- Chapter 2 details system requirements
- Chapter 3 describes how to install and configure SCO TCP/IP
- Chapter 4 shows how to start and stop SCO TCP/IP
- Chapter 5 shows how to remove SCO TCP/IP
- Chapter 6 describes how to configure SCO TCP/IP for enhanced performance on a fast computer
- Chapter 7 lists known problems and recommended solutions
- Chapter 8 lists documentation errata for the published and on-line documents that are distributed with SCO TCP/IP 1.2.1.

Conventions used in these notes

Utilities and commands are printed in **boldface** type, with the Reference Guide section in which they appear following in parentheses, for example: **cat**(C). Filenames are *italicized,* for example; */etc/passwd.* Output is printed in monospaced font, for example: `Error: incorrect volume in drive.` Text which you must key in at the terminal appears in boldface type, for example: **exit.**

RFCs are referred to throughout the TCP/IP documentation. RFCs are Requests for Comments, which are an informal, loosely coordinated set of notes on TCP/IP and the connected Internet, including its architecture, protocols and history. RFCs are available electronically from the Internet Network Information Center. The Internet domain name for the host that provides the archive is RS.INTERNIC.NET.

Chapter 1

Contents and features of this package

The following software and documentation are included in SCO TCP/IP:

- three SCO TCP/IP Runtime System diskettes
- these *Release and Installation Notes*
- *SCO TCP/IP User's and Administrator's Guide*
- *SCO TCP/IP Command Reference*

SCO TCP/IP 1.2.1 provides the following major features:

TCP	Transmission Control Protocol (RFC 793)
UDP	User Datagram Protocol (RFC 768)
IP	Internet Protocol (RFC 791)
ARP	Address Resolution Protocol (RFC 826)
ICMP	Internet Control Message Protocol (RFC 792)
PPP	Point-to-Point Protocol (RFC 1331, 1332)
RIP	Routing Information Protocol (RFC 1058)
SLIP	serial line IP STREAMS module (RFC 1055)
SNMP	Simple Network Management Protocol (RFC 1213, 1227)
NetBIOS	NetBIOS protocol (RFC 1001, 1002)
Loopback	loopback and test STREAMS module
Utilities	**rsh, rlogin, rcp, telnet** (RFC 854), **ftp** (RFC 959), **inetd** and other utilities

All features are accessible through TLI (described in the *Encyclopedia* in the SCO UNIX Development System) and XTI (as defined in the X/Open Portability Guide, Networking Services, Issue 3, Volume 7). A standard 4.3BSD UNIX socket interface is provided with the SCO TCP/IP Development System for application portability.

New features and enhancements for Release 1.2.0

SCO TCP/IP Runtime System Release 1.2.0 for the UNIX System contained many new features introduced since 1.1.3. These included:

- Berkeley **lpd** support
- **gated** support
- **snmp** agent with MIB II support
- updated **telnet** and **telnetd** for wider connectivity
- updated **sendmail** (to UCB release 5.65)
- updated SLIP support
- new dial-in/dial-out PPP
- improved performance

The following sections describe these features.

Berkeley lpd support

SCO TCP/IP now allows optional support of the standard distributed print spooling subsystem supported by Berkeley UNIX 4.3: **lpd**. Supporting the **lp**, **lpstat**, and **cancel** commands, the SCO TCP/IP **lpd** subsystem allows heterogeneous distributed printing support between SCO and other BSD **lpd** conformant systems such as SUNOS, PC-NFS, FTP's TCP, and HP-UX.

gated support

gated is an advanced routing protocol developed at Cornell University. It enables the administrator to set up the system to selectively act as an adaptive router. This router learns of changes in network topology and status and passes that information to external networks, both public and private. **gated** combines standard RIP, Hello, and EGP routing protocol support with a set of rules that constrains how it advertises routes to other gateways.

The interior gateway protocols RIP and Hello are used to collect, and act on, routing information and status from within a network. This dynamic routing information is also used in advertising routes to another network with the exterior gateway protocol EGP. The rules allow the administrator to specify which systems are advertised and to artificially raise the distance metric to control network traffic flow.

SNMP (Simple Network Management Protocol)

A standard protocol used by network management stations to monitor TCP/IP networks, SNMP defines a set of variables that gateways must keep to help monitor and control network nodes. SCO TCP/IP for UNIX Systems now includes an SNMP agent that supports a new expanded set of variables: MIB II (Management Information Base II). SCO's MIB II conforms to the RFC 1213 Internet standard.

Updated telnet and telnetd

telnet and **telnetd** are now version 91.03.25. This version improves heterogeneous connection support.

Updated sendmail

SCO TCP/IP's **sendmail** is updated to the latest release, 5.65. It takes advantage of **routed** and **gated** which allows the use of the name server for simplified configuration management. **sendmail** is a standard SCO TCP/IP mail messaging system, supporting the SCO TCP/IP standard mail exchange protocol, SMTP (Simple Mail Transport Protocol). SMTP conforms to MIL-STD 1781 and RFCs 821 and 822.

Updated SLIP support

The SLIP (Serial Line Interface Protocol) now contains Van Jacobson header compression algorithms to improve throughput and response.

PPP (Point-to-Point Protocol)

PPP enables dial-up and dial-out SCO TCP/IP connections to other TCP/IP systems. Conforming to the RFC 1171 and dialing support and Van Jacobsen header compression.

Performance improvements

A small but noticeable improvement in performance has been obtained by optimizing the code and streams utilization.

Support for additional LAN cards

SCO TCP/IP for UNIX Release 1.2.0 includes the SCO LLI Driver Disk Release 3.1, which includes support for the following LAN cards:

Driver	Vendor	Supported cards
exos	Microdyne	Excelan 205, wo5T, 205T 16-bit
e3A	3Com	3c501
e3B	3Com	3c503
e3C	3Com	3c523 3c523b
e3D	3Com	3c507
hpi	HP	ISA 2750A ThinLan PC Adapter
hpe	HP	EISA 27248A 32-bit
i3B	Racal	ES-3210
i6E	Racal	NF-6510
tok	IBM	*
wdn	WD/SMC	†

* The following IBM Token Ring adapters are supported: IBM Token Ring Adapter; IBM Token Ring Adapter II (long card), 4 MHz, for the AT; IBM Token Ring Adapter II (short card), 4 MHz, for the AT; IBM Token Ring Adapter, 4/16 MHz, for the AT; IBM Token Ring Adapter/A, 4 MHz, for the PS/2; IBM Token Ring Adapter/A, 4/16 MHz, for the PS/2.

† The following Western Digital cards are supported: WD8003E, WD8003EBT, WD8003E/A, WD8003ET/A, WD8003EP/A, WD8003W/A, WD8003WT, WD8003W, WD8003EB, WD8003EP, WD8013EBT, WD8013EP, WD8013W, WD8013EW, WD8013EPC, WD8013WC, WD8013EWC, WD8013EPA, WD8013WPA.

New features and enhancements for Release 1.2.1

SCO TCP/IP Runtime System Release 1.2.1 for the UNIX System contains the following enhancements over previous releases:

- support for SCO UNIX Operating System enhancements
- updated PPP (Point-to-Point Protocol) package
- support for additional LAN cards
- updated NetBIOS package
- updated SNMP (Simple Network Management Protocol) package

The following sections describe these features.

Support for operating system enhancements

SCO TCP/IP 1.2.1 includes support that allows more than 256 TCP connections at a time using the SCO Extended Minor Device numbering scheme (refer to the **hd**(HW) and **mdevice**(F) manual pages for more information). The number of active TCP/IP connections allowed per system is configurable in lots of 256 connections via the **netconfig** utility during TCP/IP installation.

SCO TCP/IP 1.2.1 also allows networking applications to open up to 11,000 file descripters at one time. Previously, this limit was 150 open files.

Updated PPP (Point-to-Point Protocol) package

PPP is implemented to conform to the latest Internet standards: RFC 1331, RFC 1332, and part of Internet Draft "The PPP Authentication Protocols" by B. Lloyd and W. Simpson. The major changes resulting from implementation of the new RFCs bring improved reliability through use of a new state table design and increased interoperability through an increase in negotiable parameters. From the Internet Draft, this PPP implements the Password Authentication Protocol which provides a system ID and password authentication scheme which can be used as a security prerequisite to connection establishment. In addition, a new user command, **pppstat**(ADMN), has been added which retrieves and displays PPP connection status. For more information see the "Documentation errata" section later in these notes.

Updated NetBIOS package

The NetBIOS module (TPI NetBIOS) provided in this release is a replacement for the NetBIOS module shipped with SCO TCP/IP 1.2.0. The NetBIOS module shipped with SCO TCP/IP 1.2.0 supported only the DO_NCB interface. The TPI NetBIOS module continues to support the DO_NCB interface. Additionally it supports the Transport Provider Interface (TPI). All other functionality remains the same.

Application programmers can access the NetBIOS transport through the SCO TLI and XTI libraries. For further information on these libraries, see the *Encyclopedia* in your Development System documentation set.

> **NOTE** The programmer who is using the XTI or TLI library to develop an application must specify, via the **t_open**(NSL) routine, the type of service desired. To access the connection oriented service of the NetBIOS transport, open the device */dev/nbcots*. To access the connectionless service of the NetBIOS transport, open the device */dev/nbclts*.

The new Microsoft LAN Manager 2.2 Server for SCO UNIX System V Release 3.2 Version 4 requires a NetBIOS module that supports the TPI. The older Microsoft LAN Manager 1.1.0 for UNIX systems requires DO_NCB interface support. The multiple interface support of the TPI NetBIOS module makes this module compatible with LAN Manager versions 1.1.0, 2.1 and 2.2.

Updated SNMP (Simple Network Management Protocol) package

SNMP is an implementation of the Simple Network Management Protocol. This implementation supports all the objects defined in MIB-II (see RFC 1213). New with this release of SNMP is an implementation of the SMUX (SNMP Multiplexing) protocol. This protocol allows a user-process to act as a sub-agent (referred to as an "SMUX peer") and export a portion of the MIB to the local SNMP agent. This serves to enhance the number of objects that can be managed by the local SNMP agent without modifying it. Additional information on SMUX is provided in the "Documentation errata" section, later in these notes.

Chapter 2

System requirements

The following chart details the resources needed to run SCO TCP/IP:

Computer	industry standard (AT), Microchannel, or EISA computers with 80386 or 80486 processor chips
Disk capacity	at least 3835 Kbytes (3.75 Mbytes) of hard disk storage for SCO TCP/IP Runtime System under the UNIX system
RAM	at least 8 Mbytes of RAM for SCO TCP/IP Runtime System and SCO UNIX systems
Operating system	SCO UNIX Release 3.2 Version 4.2 or Release 3.2 Version 4 with SCO UNIX Maintenance Supplement v4.2
Software	SCO NFS Release 1.2.1, SCO LLI Release 3.1 and Link Kit (LINK) packages: and the operating system extended utilities Netconfig (NETCFG) package and optionally the operating system's **mail** (MAIL) and manpages

Supported networking hardware

SCO TCP/IP Runtime System officially supports all drivers found in the SCO LLI Driver Disk Release 3.1, including standard and Microchannel Ethernet cards, SLIP and PPP lines, and Token Ring adapters.

The SLIP and PPP protocol drivers provided with SCO TCP/IP use the regular SCO tty device interface and work with any serial card supported by the SCO SIO (serial I/O) driver. The Release Notes for your operating system contain a list of serial I/O boards supported by the SIO driver.

For more information on officially supported cards, please refer to the *SCO LLI Driver Disk Release and Installation Notes.*

Chapter 3

Installing and configuring SCO TCP/IP

This chapter provides an overview of SCO TCP/IP installation and configuration, the installation procedure, and information on how to configure SCO TCP/IP. If you have never installed SCO TCP/IP or other networking software, we strongly advise that you read Chapter 1, "Networking and TCP/IP overview," of the *SCO TCP/IP User's and Administrator's Guide* before proceeding.

Preparing for SCO TCP/IP installation

Before installing SCO TCP/IP, be sure that the following conditions are met:

- The Link Kit and **netconfig** must be installed. For details, refer to the *SCO UNIX Operating System Installation Guide.*
- SCO UNIX Release 3.2 Version 4.2 or Release 3.2 Version 4 with SCO UNIX Maintenance Supplement v4.2 must be installed. For details on installing or updating the Maintenance Supplement, see the *SCO UNIX Operating System Maintenance Supplement Notes.*
- If SCO MPX 2.0 is installed on your system, it will be removed on installation of Maintenance Supplement v4.2 for SCO UNIX. You will then be asked if you wish to install SCO MPX 3.0.
- Make sure the SCO LLI Driver Disk Release 3.1 is available for installation. It is correct to install it after installing TCP/IP as shown in the steps in the next section, "Overview." For details, see the *SCO LLI Driver Disk Release and Installation Notes.*
- Earlier versions of SCO TCP/IP will be upgraded when you install SCO TCP/IP Release 1.2.1.

Overview

After you prepare your system for SCO TCP/IP, you are ready to install the software. The following procedure is a general overview of installing and configuring SCO TCP/IP Runtime System. It refers you to subsequent sections of these notes and to other guides for more information.

1. Log in as *root* and bring the system to single-user mode.
2. Be certain that you are in the *root* (/) directory before you start the installation procedure.
3. Install TCP/IP software using the **custom** utility.
4. Install SCO LLI software using the **custom** utility.
5. Create an installation worksheet listing the type of network cards you are installing and the values to enter during network configuration. This information is listed in the driver-specific checklists in the *SCO LLI Driver Disk Release and Installation Notes* and in the section, "Answering netconfig prompts" later in these notes.
6. Configure your Ethernet boards, SLIP or PPP connections, or Token Ring adapters using the **netconfig** utility.
7. Test TCP/IP, as shown in Chapter 7, "Network administration," of the *TCP/IP User's and Administrator's Guide.*
8. Enable additional desired features, such as SNMP or remote line printing, by following the procedures described in the *TCP/IP User's and Administrator's Guide.*
9. Adjust the MMDF or **sendmail** configuration files for your installation.
10. Tune default configurations, if needed. For a discussion of tunable parameters, see Chapter 7, "Network administration," of the *TCP/IP User's and Administrator's Guide.*

Installing TCP/IP Runtime System

Under the SCO UNIX system, installation is done through the **custom** utility. Follow the steps in this section to install TCP/IP Runtime System.

> **NOTE** The operating system's Link Kit package must be installed before TCP/IP can be installed. The operating system's Netconfig package must be installed before you can configure TCP/IP. In addition, the MAIL package of the operating system must be installed if you want to use **sendmail** as provided with TCP/IP.

When you upgrade TCP/IP, the upgrade attempts to retain all compatible configuration files. The table below lists the files that are retained in place.

Networking configuration files saved in place

/etc/exports
/etc/hosts
/etc/hosts.lpd
/etc/printcap
/etc/named.boot

Any incompatible files are backed up in the backup directory with the suffix *.old*. The table below lists these files and their backup directories.

Incompatible networking configuration files

Configuration file	Backed up after upgrade
/etc/nfs	/install/nfsrt_upgrade/etc/nfs.old
/etc/ethers	/install/tcprt_upgrade/etc/ethers.old
/etc/inetd.cons	/install/tcprt_upgrade/etc/inetd.conf.old
/etc/protocols	/install/tcprt_upgrade/etc/protocols.old
/etc/services	/install/tcprt_upgrade/etc/services.old
/etc/sockcf	/install/tcprt_upgrade/etc/sockcf.old
/etc/strcf	/install/tcprt_upgrade/etc/strcf.old
/etc/tcp	/install/tcprt_upgrade/etc/tcp.old
/etc/snmp	/install/tcprt_upgrade/etc/snmp.old
/etc/default/tcp	/install/tcprt_upgrade/etc/default/tcp.old
/etc/default/nbconf	/install/tcprt_upgrade/etc/default/nbconf.old
/etc/systemid	/install/tcprt_upgrade/etc/systemid.old
/usr/lib/uucp/systems	/install/tcprt_upgrade/usr/lib/uucp/systems.old
/usr/lib/named/named.soa	/install/tcprt_upgrade/usr/lib/named/named.soa.old
/usr/lib/named/named.local	/install/tcprt_upgrade/usr/lib/named/named.local.old
/usr/lib/named/named.hosts	/install/tcprt_upgrade/usr/lib/named/named.hosts.old
/usr/lib/named/named.new	/install/tcprt_upgrade/usr/lib/named/named.new.old
/usr/lib/named/root.cache	/install/tcprt_upgrade/usr/lib/named/root.cache.old

The new (mainly empty) configuration files (which are backed up because the previously configured files are used instead) are put into the backup directory with the suffix *.new*. The table below lists these files and their backup directories.

Backed up new networking configuration files

Configuration file	Backed up new file after upgrade
/etc/bootptab	/install/tcprt_upgrade/etc/bootptab.new
/etc/hosts.equiv	/install/tcprt_upgrade/etc/hosts.equiv.new
/etc/networks	/install/tcprt_upgrade/etc/networks.new
/etc/shells	/install/tcprt_upgrade/etc/shells.new
/etc/snmpd.conf	/install/tcprt_upgrade/etc/snmpd.conf.new
/etc/snmpd.comm	/install/tcprt_upgrade/etc/snmpd.comm.new
/etc/snmpd.trap	/install/tcprt_upgrade/etc/snmpd.trap.new
/etc/snmpd.peers	/install/tcprt_upgrade/etc/snmpd.peers.new
/etc/if.ignore	/install/tcprt_upgrade/etc/if.ignore.new

To install TCP/IP, follow these steps:

1. Bring the system to system maintenance mode by entering **init 1** from the command line on the first console multiscreen, *tty01*. After several shutdown messages, the following prompt appears:

   ```
   Type CONTROL-d to proceed with normal startup,
   (or give root password for system maintenance).
   ```

   Enter the *root* password.
2. Type **custom** and press ⟨Return⟩.
3. The initial **custom** menu appears. Select Install.
4. You are prompted to select a product to install. Choose A New Product (even if you have a previous version already installed) and press ⟨Return⟩.
5. The Install menu appears. Select Entire Product to install all of the packages, or select Packages to install a subset of the product. Unless you are updating individual packages or have space constraints on your hard disk, select Entire Product. If you are upgrading the following message appears:

   ```
   Upgrading SCO TCP/IP Runtime System rel=1.2.0x to release rel=1.2.1x
   ```

   (the last characters on the line will be truncated). You are asked if you wish to continue. Select Yes to continue. A series of upgrade messages appear.

6. You are prompted to insert distribution volume 1. Insert the floppy into drive 0 and press ⟨Return⟩ to continue. The following messages appear:

   ```
   Installing Custom Data Files...
   Creating file lists...
   ```

7. If you selected Packages, a display of package names appears. Select the desired packages by moving the cursor to each package, then pressing the space bar. When all desired packages are marked, press ⟨Return⟩. The disks you are prompted to insert depend on the packages you chose.

   > **NOTE** If you select ALL, MMDF becomes the active mailer. The active mailer is not **sendmail** unless explicitly configured, as described in the chapter "TCP/IP sendmail administration" of the *TCP/IP User's and Administrator's Guide.*

8. You are prompted to insert TCP/IP volume 1. Because it is already in the floppy disk drive, press ⟨Return⟩ to continue. The following message appears:

   ```
   Extracting files...
   ```

   It takes several minutes for the files to be extracted from volume 1.

9. You are prompted to insert volume 2. Insert the volume and press ⟨Return⟩ to continue. The file extraction message reappears.

10. You are prompted to insert volume 3. Insert the volume and press ⟨Return⟩ to continue. The file extraction message reappears.

11. After a few moments, the restricted rights legend appears, and the following message prompts you for your serial number:

    ```
    Enter your serial number or enter q to quit:
    ```

    Enter the serial number as it appears on your TCP/IP serial number and activation key card and press ⟨Return⟩.

12. The following message prompts you for your activation key:

```
Enter your activation key or enter q to quit:
```

    Enter the activation key as it appears on your TCP/IP serial number and activation key card and press ⟨Return⟩.

    You can remove volume 3 at this point.

13. Over the next few minutes, several messages detail the installation. If you are upgrading, upgrade messages appear. Then these messages appear:

```
Changing streams resources needed for TCP/IP... done

Installing drivers into the link kit... done

Enter the system node name or enter q to quit [scosysv]:
```

    The default machine name appears in brackets; to accept it, press ⟨Return⟩. Otherwise, enter the name of your machine and press ⟨Return⟩.

14. The following prompt appears:

```
Enter DOMAIN name for machine or enter q to quit [machine.UUCP]:
```

    The default domain name appears in brackets; to accept it, press ⟨Return⟩. Otherwise, enter the domain name and press ⟨Return⟩.

15. The following message and prompt appears:

```
256 TCP connections currently configured, do you want to:

          1. Add TCP Connections
          2. Remove TCP Connections

Select an option or enter q to quit [q]:
```

    If you decide to add or remove TCP connections, you must do so in multiples of 256.

    After adding or removing TCP connections or entering "q", the system informs you that TCP/IP configuration is complete.

16. You are asked if you want to relink the kernel.

    NOTE To save time, do not relink the kernel until you have used **netconfig** to configure your network.

    If you type **n**, you see the message:

    ```
    SCO TCP/IP Runtime installation is complete
    Please use netconfig to configure SCO TCP/IP
    ```

    Press any key to continue and the message "Checking file permissions..." appears, followed by a short pause. Skip to step 18.

    If you type **y**, the following messages appear:

    ```
    The UNIX Operating System will now be rebuilt.
    This will take a few minutes, please wait.
    Root for the system build is /.
    ```

    After a pause, this is followed by:

    ```
    The UNIX kernel has been rebuilt.
    Do you want this kernel to boot by default?
    ```

    If you type **y**, your old kernel is copied to */unix.old*, and your new kernel is copied to */unix*.

17. If you relinked your kernel, you see the messages:

    ```
    The kernel environment includes device node files
    and /etc/inittab. The new kernel can require
    changes to /etc/inittab or device nodes.
    Do you want the kernel environment rebuilt (y/n)?
    ```

    NOTE You must rebuild the kernel environment to run TCP/IP. You should not answer **n** to this prompt.

If you type **y**, you see the messages:

```
The kernel has been successfully linked and installed.
To activate it, reboot your system.

Setting up new kernel environment...
SCO TCP/IP Runtime installation complete.
Please use netconfig to configure SCO TCP/IP
```

Press any key to continue. The message "Checking file permissions ..." appears for a short time. Skip to step 18.

If you type **n**, you see the message:

```
Device node or inittab changes associated with this
new kernel have not been made. These changes should
be made by running: touch /etc/.new_unix;
/etc/conf/bin/idmkenv.
```

18. The main **custom** menu is displayed. Select **quit** and press ⟨Return⟩. Select yes and press ⟨Return⟩ to exit. TCP/IP is now installed on your system.
19. Configure SCO TCP/IP Runtime System, including drivers, by running the **netconfig** program. This process is described in the next section "Configuring TCP/IP."

Configuring TCP/IP

Use the **netconfig**(ADM) utility to configure TCP/IP (and other SCO networking products) by entering **netconfig** on the command line while in single-user mode. For more information on **netconfig**, see its manual page.

> **NOTE** Before you configure TCP/IP with **netconfig**, you must install the SCO LLI Driver Disk as described in the *SCO LLI Driver Disk Release and Installation Notes*. If you do not, you will not have access to any networking drivers other than the loopback, SLIP and PPP drivers.

When you use **netconfig**, you build *chains* that link applications software to transport software and networking drivers. A chain is a simplified way of looking at how different networking software programs and hardware interact. Your networking software does not work until you create the appropriate chains.

For example, if you wanted to configure NFS over TCP/IP over a 3COM 3c523 (e3C) driver, you would create the following chains:

```
sco_tcp->e3C0
nfs->sco_tcp
```

Each time you create a chain, the **netconfig** command prompts you for information on the software or hardware you are configuring. The checklists found in the *SCO LLI Driver Disk Release and Installation Notes* and the networking overview found in the *TCP/IP User's and Administrator's Guide* help you answer these prompts.

The following steps describe a general overview of adding a chain with **netconfig**.

> **NOTE** You should run **netconfig** as *root*, in single-user mode, because **netconfig** often requires you to relink the kernel.

1. Access **netconfig** from your operating system prompt. The main menu appears.
2. Select Add a Chain. A list of configurable software appears.
3. Select the software you want to configure.

   > **NOTE** You must configure TCP/IP before configuring LAN Manager Client.

4. Continue configuring chain elements until you reach the lowest level of the chain. After you make the final selection, you are prompted to enter information about the chain.

   > **NOTE** If you enter an IRQ, DMA, RAM address, or ROM address that is already in use by another device on your system, you can either pick another value, or take the desired value away from the other device. If you choose the latter, **you must reconfigure the other device before you relink the kernel**, so that device will continue to function on your system. We recommend that you use the second option with extreme care.

5. After configuring the chain, you can add another chain, reconfigure a chain, remove a chain, or exit the main menu.
6. When you exit **netconfig**, you are sometimes prompted to relink the kernel. If you are, you must relink for the changes you made to take place. Type **y** at each of the relinking prompts to relink the kernel successfully, then reboot your system.

Sample netconfig sessions

The following examples describe the use of **netconfig** in specific terms. Your specific **netconfig** sessions will probably be different.

> **NOTE** We recommend that you run **netconfig** as *root*, in single-user mode, because **netconfig** often requires you to relink the kernel.

In this first example, TCP/IP is configured over a 3COM 3c523 driver:

1. Enter **netconfig** from the operating system prompt.

   The main **netconfig** menu appears, from which you can add, remove, or reconfigure chains. You make menu choices by entering the number that corresponds to the desired choice.

2. Select Add a New Chain.

   A list of available software appears. This list contains all possible top levels of chains, based on what is currently installed on your system.

3. Select sco_tcp.

   A list of available drivers appears. Each driver corresponds to a particular networking card or SLIP or PPP line. Your computer may have one or more types of networking cards available.

4. Select e3C0.

   This choice corresponds to the 3COM 3c523 driver.

5. Confirm the chain by entering **y** at the confirmation prompt.

6. You now begin the actual configuration process, where you must enter information on both the TCP/IP software and the 3COM card. The section "Answering netconfig prompts," found later in these notes, helps this process.

   When done, you return to the main **netconfig** menu.

7. If you have finished configuring chains, select Quit. You are prompted to relink the kernel. Do so, or your changes will not go into effect.

   If you want to configure another chain, select Add a New Chain and choose a new top level.

In the next example, NFS is configured to run over TCP/IP.

NOTE TCP/IP must already be configured before you configure NFS.

1. Enter **netconfig** from the operating system prompt.

   The main **netconfig** menu appears.
2. Select Add a New Chain.
3. Select nfs.
4. Select sco_tcp.
5. If you have finished configuring chains, select Quit. Type **y** at the relink prompts to relink your kernel.

   If you want to add more chains, select Add a New Chain.

Answering netconfig prompts

When you use **netconfig** to configure TCP/IP, you are actually configuring both TCP/IP and a driver in the SCO LLI Driver Disk. Driver-specific prompts are discussed in the *SCO LLI Driver Disk Release and Installation Notes.*

Prompts specific to using TCP/IP over a particular driver are:

- Internet address — an address that uniquely identifies your machine on the network. In the case of a machine with multiple networking cards or serial lines, each driver must have its own Internet address.

  NOTE If you enter an Internet address or hostname that already appears in */etc/hosts*, a warning message appears, telling you to resolve any possible conflicts by editing */etc/hosts*. See the *hosts*(SFF) manual page for more information on legal file formats.

- Netmask — a value that masks the Internet portion of your Internet address, leaving only the host portion.
- Broadcast address — an address that TCP/IP uses to broadcast packets to the entire network, rather than to a specific destination.
- Local node name — a name you specify for the interface you are configuring.
- Gateway status — whether your machine serves as a gateway between two networks; only valid in the case of a multiple-card or multiple-serial line system.
- Number of pseudo ttys — the number of pseudo ttys allocated to network programs, such as **telnet** or **rlogin**.

  NOTE You must reserve a minimum of 16 pseudo ttys for SCO TCP/IP.

- Number of TCP/IP connections — the number of active TCP/IP connections allowed per system.

More detailed information on these prompts, as well as a background discussion of configuring network drivers for TCP/IP, is found in Chapter 1, "Networking and TCP/IP overview," and Chapter 7, "Network administration," of the *TCP/IP User's and Administrator's Guide.*

Reconfiguring TCP/IP

To change information about a previously configured chain, use **netconfig**. The procedure for reconfiguring a chain element (such as a driver) is the same as for the initial configuration, with one exception: instead of choosing "Add a chain," choose "Reconfigure a chain."

> **NOTE** If you remove all of the chains that another product, such as NFS, runs over, you may need to remove the NFS chain, then add it back after reconfiguring the chains on which it relies.

Configuring NetBIOS

To be used, NetBIOS must be configured into a chain of network products using **netconfig**. Configuring NetBIOS is a process very similar to those found in the examples in the "Sample netconfig sessions" section found earlier in these notes. The main difference, as in the configuration of any network product, is in the NetBIOS specific items for which **netconfig** prompts you.

NetBIOS specific netconfig prompts

Prompts specific to using NetBIOS are:

- Network host name — name of machine on which NetBIOS is running.
- NetBIOS Scope Identifier — domain name of the network over which your system will be running NetBIOS. An example domain name is "sco.com." For more information on domain names, see your *SCO TCP/IP User's and Administrator's Guide.*

Reconfiguring and deconfiguring NetBIOS

To change information about a chain, or remove a chain, of which NetBIOS is a part, use **netconfig**. The procedure to change (i.e., reconfigure) a chain element or remove (i.e., deconfigure) a chain is the same as the initial configuration, with this exception: choose "Reconfigure a chain" or "Remove a chain," as appropriate, from the **netconfig** menu.

Chapter 4

Starting and stopping TCP/IP

TCP/IP starts automatically when your system enters multiuser mode and stops when you enter single-user mode. If you want to start or stop TCP/IP manually while in multiuser mode, use the **tcp stop** and **tcp start** commands.

To halt the TCP/IP software, type:

```
/etc/tcp stop
```

To restart TCP/IP, type:

```
/etc/tcp start
```

If you are running in "high" security mode, you will need to type:

```
sd tcp stop
```

to halt the TCP/IP software, and:

```
sd tcp start
```

to restart TCP/IP.

> **NOTE** If NFS or NetBIOS are running over TCP/IP, you need to stop those products before stopping TCP/IP. Refer to the documentation provided with NFS for information on stopping this product from the command line. NetBIOS must be deconfigured. See the section "Configuring NetBIOS" earlier in these notes.

If there is an activation state mismatch between TCP/IP and the underlying operating system, the following message appears at the system console at each invocation of the **/etc/tcp** start command:

```
WARNING: Activation state mismatch.
      TCP Startup proceeding
```

Chapter 5

Removing SCO TCP/IP Runtime System

Because TCP/IP is an integral part of your system and is linked with your operating system kernel, removal of the software is a delicate and important task. Before removing TCP/IP, you should remove all other software, such as SCO NFS, that relies on TCP/IP. It is crucial that you follow these instructions closely. Read through the instructions completely before you begin. Follow these steps to remove SCO TCP/IP Runtime System from your UNIX system:

1. Log in to the system as *root* and bring the system to system maintenance mode.
2. Enter **custom** at the prompt.
3. Select Remove.
4. Select SCO TCP/IP Runtime System from the list of installed software.
5. Select ALL to remove the entire product, or choose packages to remove from the point-and-pick list by moving the cursor to the package to remove, then pressing the space bar. When you have marked all of the packages you want to remove, press ⟨Return⟩.
6. After several messages asking you to verify the removal and telling you that the files and drivers are being removed, you return to the **custom** menu.
7. Select **quit** and press ⟨Return⟩ to leave **custom**. Select **yes** and press ⟨Return⟩ to exit.
8. You must now relink the kernel. Enter the following commands:

   ```
   cd /etc/conf/cf.d
   ./link_unix
   ```

   Answer yes at each of the prompts.
9. Reboot the system with the **init 6** command.

Chapter 6

Configuring TCP/IP for enhanced performance

If you have a fast (16- or 32-bit) computer with a large amount (16 Mbytes or greater) of RAM, you may see considerable performance improvement by altering two variables in the file */etc/conf/pack.d/tcp/space.c*. The following table shows the variables, their default values, and their desired values.

Variable	Default value	Desired value
TCPWINDOW	4*1024	24*1024
tcp_round_mss	1	0

To change these variables, follow this procedure:

1. Log onto your systems as *root*.
2. Change directories to */etc/conf/pack.d/tcp*.
3. Back up the file *space.c* by entering the following command:

   cp space.c space.c.old
4. Edit *space.c* to contain the desired variable values.
5. Change directories to */etc/conf/cf.d*.
6. Relink the kernel by entering the following command:

   ./link_unix

   Enter **y** at each of the relink prompts.
7. Reboot your computer to load the new kernel.

This configuration has been shown to increase throughput on fast computers. If there are any problems with this new configuration on your computer, you can restore the default behavior by moving the backup file *space.c.old* to its original location, *space.c,* then relinking the kernel and rebooting your computer.

Chapter 7

Known limitations with this release

This section contains information on software and hardware limitations, unsupported features, and workarounds that you may need when you install and use TCP/IP.

- If you installed a previous version of TCP/IP, you must follow the instructions in the *Release and Installation Notes* of that software version to remove it before installing the latest version of TCP/IP or use the upgrade path provided if you are upgrading from TCP/IP Release 1.2.0 to TCP/IP Release 1.2.1.
- If you are removing TCP/IP Release 1.1.3f from a SCO UNIX Release 3.2 Version 4 system, part of the operating system's mail package is also removed. You must re-install the operating system's mail package before installing TCP/IP Release 1.2.0. You are also prompted for the UA2 floppy of the Maintenance Supplement Version 4.1 when installing TCP/IP 1.2.0.
- The **rlogin** command does not always successfully pass the **TERM** environment variable to the remote system. To ensure that it does, users must make a minor change to their *.profile* or *.cshrc* files.

  sh users edit *.profile* to replace this line:

  ```
  eval `tset -m ansi:ansi -m :\?ansi -e -s -Q`
  ```

  with this line:

  ```
  eval `tset -m ansi:ansi -m :\?$TERM -e -s -Q`
  ```

  ksh users edit *.profile* to replace this line:

  ```
  eval `tset -m ansi:ansi -m $TERM:\?${TERM:-ansi} -r -s -Q`
  ```

  with this line:

  ```
  eval `tset -m ansi:ansi -m :\?${TERM:-ansi} -r -s -Q`
  ```

csh users edit *.cshrc* to replace this line:

```
term = ('tset -m ansi:ansi -m :?ansi -r -S -Q')
```

with this line:

```
term = ('tset -m ansi:ansi -m :?$TERM -r -S -Q')
```

- The **tftp** command has been configured, by default, not to operate under the UNIX System due to security considerations. This command, used in an uncontrolled setting, may create security holes in your system. However, using the command with the **-s** option ensures secure operation of this command.

  To enable **tftp**, uncomment one of the following lines in */etc/inetd.conf* and reboot your system:

  ```
  #tftp   dgram   udp   wait   nouser    /etc/tftpd   tftpd
  #tftp   dgram   udp   wait   root      /etc/tftpd   tftpd -s /tftpboot
  ```

  If you uncomment the first line, you enable **tftp** in non-secure mode; the second line enables **tftp** in secure mode. If you use the **-s** option, you must create the directory */tftpboot* manually.

  Any changes you make to */etc/inetd.conf* take effect the next time TCP/IP starts. For more information on the command, the command's daemon, and associated files, refer to the **tftp** and **tftpd** manual pages.

- The **rwhod** daemon is commented out of the */etc/tcp* startup script, as it can drain system performance. If you want to use **rwhod**, uncomment the following three lines:

  ```
  #       if [ -x /etc/rwhod -a -d /usr/spool/rwho ]; then
  #               rwhod ; echo "rwhod \c"
  #       fi
  ```

- Some character loss has been reported when using the operating system's **vi** command over **rlogin**. This is due to a missing or incorrectly set **tab3** terminal environment variable on the remote terminal session. To correct this problem, enter the following command after you log into the remote system:

  stty tab3

- You cannot configure SLIP lines to a non-default netmask setting with **netconfig**. This is the desired behavior. However, sites with special needs, such as a site that consists of subnets, may edit */etc/tcp* to contain the correct netmask. To do so, move to the line near the top of the file that contains `NETMASK=` and add the desired netmask.

- Occasionally, transferring a large file with **ftp** between a system running TCP/IP for SCO UNIX Release 1.2.0 and TCP/IP for SCO UNIX Release 1.1.3 fails with the error message:

  ```
  425 Can't build data connection: address already in use
  ```

  In this case, issue **ftp**'s **sendport** command, then transfer the file again.

- The IBM PS/2 386SX MC does not correctly recognize the Western Digital WD8003E/A network card. This is a hardware problem unrelated to TCP/IP software.

- **sendmail** does not handle delivery through Micnet or multiple UUCP gateways. It does support single UUCP gateways, local mail, and mail sent through TCP/IP.

- If you use **sendmail** on a machine that does not use **named**, users must enter fully-qualified mail addresses, including the correct domain name. For example, the address *john@toaster* is invalid, while *john@toaster.UUCP* is a valid address.

- The **mkdev slip** command is obsolete. Use **netconfig** to configure network interfaces, including SLIP lines.

- The operating system can hang during serial device initialization if a Western Digital card is present. If this occurs, you must change the base I/O address in the TCP software and on the Western Digital card. Addresses 240 and 380 have been tested successfully. To reconfigure the software, use the **netconfig** utility and reconfigure the appropriate chain.

- A large number of unreferenced files can appear on the *root* filesystem, even after a clean system shutdown. This is due to the behavior of the STREAMS cloning driver. This can usually be avoided by shutting down TCP/IP before shutting down the system. The system administrator should also periodically run **fsck**(ADM) while in system maintenance mode to clean up the filesystem. This should usually be done about once a week, depending on whether the system is brought up and down frequently and whether network usage is heavy or light.

- Due to the interaction between STREAMS, VP/ix, and the kernel, network performance is impaired when running VP/ix.

- Using **rlogin** or **telnet** to log in to a machine running SCO UNIX Release 3.2 can give the error message `bad login user id`. This error does not occur when the destination machine is running SCO UNIX Release 3.2 Version 4 in either of the two lower security modes ("traditional" and "low").

This message appears when someone else logs in as *root* and does a **tcp stop** followed by a **tcp start**. Because the UNIX system has C2 security, the login ID is automatically set when someone logs in as *root*. This login ID is passed to **inetd**, which passes it to **rlogind** or **telnetd**. **rlogind** or **telnetd** use **/bin/login**, which inherits the previously set login ID (which was *root*). Then the login procedure calls **setluid( )**, which fails because the login ID is already set.

To avoid this error, start TCP/IP using the **sd**(ADM) command while logged in as *root*. To do so, enter the following command:

```
sd tcp start
```

The **sd** command starts other commands without an LUID. Only *root* and other users with the **sysadmin** authorization are permitted to run **sd**.

- The **netstat** command may not always work properly under load. Some commands (such as **netstat -a** or **netstat**) can give the error message:

  ```
  corrupt control block chain
  ```

  Try using the **llistat**(ADM) command if you encounter this error.

- If you use application programs over the network, using the direction keys to move the cursor can add unwanted text or commands to the application. This is due to the packet nature of the network transmissions. Direction keys generally send a multiple character sequence to affect cursor motion. If this sequence is broken across packet lines, your application may not interpret the character sequences correctly. Some applications, such as **vi**, can be configured to wait for multiple-character sequences. Check the documentation of these applications to determine whether or not this capability is offered.

- While interrupt vector 2 is valid on all machines with TCP/IP, interrupts on IRQ2 are sometimes lost because of inconsistencies in some industry standard computers. This is a hardware problem, not a problem with TCP/IP. If you certified that your system is correctly installed but you are losing interrupts on IRQ2, it is due to a hardware defect. To work around this problem, choose another available interrupt vector and reconfigure your software (using **netconfig**) and hardware (by changing jumper settings or using a setup program, if needed).

- An operating system with a two-user license will deny more than one **telnet** login if any user is logged into the console. A two-user system is limited to the following scenarios:

  - unlimited console logins
  - unlimited console logins and one **telnet** login
  - two **telnet** logins with no console logins

This is the correct behavior. Contact SCO's Sales department for information on upgrading to an unlimited license.

- Parsing of */etc/sockcf* causes the default protocol within a type to be the last specification instead of the first. This is caused when the kernel constructs the protocol table by adding new entries to the beginning instead of the end.

  To workaround this problem, reverse the order of the entries in */etc/sockcf* for protocols of the affected type(s). You can correct the problem by changing the kernel to add new entries to the end of the table instead of the beginning.

- The TCP/IP command **rcp** has a different functionality than that of the **rcp** supplied with the UNIX system. The UNIX **rcp** spools up requests for files to be copied across a Micnet network, while the TCP/IP **rcp** performs an immediate file copy across a TCP/IP Internet. Neither command knows about the other, and they do not cooperate in any way. Also, they support different sets of command line options. When you install TCP/IP, the UNIX **rcp** is moved to */usr/lib/custom/save/rcp* and the TCP/IP **rcp** is installed in */usr/bin/rcp*. The user must follow appropriate procedures to invoke the desired version of the command to access the appropriate network. If you remove TCP/IP with the **custom** utility, the original **rcp** moves back to its default location.

- When the **rcmd** command is executed from a system running the UNIX operating system to a remote system running the XENIX operating system, the connection will fail if the account on the XENIX system is set up without a password.

- There is an incompatibility between **bootp** and **tftpd** when **tftpd** is operated in secure mode.

  When a client requests a bootfile, they send a **BOOTREQUEST** asking for the file */tftpboot/bootfile* (for example). The **bootp** on the host processes this, tests for the existence of the file and sends a reply containing the validated path */tftpboot/bootfile*. A tftp request is then sent to get the required file. If, for example, **tftpd** on the host has been invoked by **tftpd -s tftpboot**, then when the request is received, the secure directory path will be prepended to the pathname in the download request to generate */tftpboot/tftpboot/bootfile*, which does not exist.

  To workaround this problem, you should have a dummy bootfile in / so that a request to download */bootfile* would return a valid result, and then **tftpd** would respond by sending */tftpboot/bootfile*.

- You may have problems with network communication via PPP if the machine running PPP meets the following conditions:
  - it has more than one serial line configured for PPP communications
  - it has PPP configured with IP address negotiation turned off (default configuration has IP address negotiation turned off).
  - it is the machine accepting the PPP call

  To resolve this problem, turn on IP address negotiation. If you cannot use IP address negotiation, configure only one serial line for use with PPP.

  Evidence of failure is not obvious, because PPP will not report errors, even though communication between IP drivers has failed. You can detect the error between two machines, 'A' and 'B', by executing **ping B** from machine A (where B is the machine accepting the call and A is the machine placing the call.)

  If the ping fails, this problem may have occurred. To obtain further evidence, run the command **netstat -i** on machine B and look at the output. If this problem is happening, then the **netstat** output will show that, for the PPP interface, the number of incoming packets increases for each ping but the number of outgoing packets remains at zero.

- The user may occasionally get a message "No route to host" when trying to connect to another peer while PAP is turned ON in */etc/ppphosts.* **cu** will work, but **ping**, **telnet**, and **rlogin** will not always work. For example, when **cu** accesses a configured PPP port, the next **rlogin** will fail, then **telnet** will fail, and so will **ping**. If PAP is turned OFF this will minimise the problem.

  However, when you use **netconfig** to turn off PAP, you must make sure that you back up */etc/ppphosts* first. Turning off PAP overwrites PPP configurations in */etc/ppphosts* with standard configurations, losing all custom configurations for that device.

- ipaddr should be turned on for systems with multiple PPP configurations.
- If using LAN Manager over NetBIOS, then you should make sure you stop the LAN Manager services before the NetBIOS services. If this is not done, then the system may panic.

Chapter 8

Documentation errata

The following section defines known errata in the printed and online documentation set.

Updated and new online manual pages

The following online manual pages are either new for SCO TCP/IP 1.2.1 (and have no corresponding page among the printed manual pages) or have been updated online for SCO TCP/IP 1.2.1 (while the printed copies have not been updated): **ftp**(TC), **ftpd**(ADMN), **icmp**(ADMP), **nbd**(ADMN), **nbtpi**(ADMP), **netbios**(ADMN), **netbios**(ADMP), **nslookup**(TC), **ppp**(ADMN), **ppp**(ADMP), **pppd**(ADMN) **pppstat**(ADMN), *ppphosts*(SFF), *pppauth*(SFF,) **route**(ADMN) **routed**(ADMN) **rshd**(ADMN,) **slip**(ADMP), **snmpd**(ADMN), *snmpd.comm*(SFF) *snmpd.peers*(SFF), **sock**(ADMP), and **telnet**(TC).

ftpd(ADMN)

When ~ftp is owned by user *ftp* this allows a breach of system security in the *FTP* directory. The directory should be owned by *root* and unwritable by anyone.

The current "Files" section states that "If your */bin/ls* is linked with shared libraries, you will need to copy */shlib/libc_s*.". Note that **ls** is compiled with shared libraries as the default.

telnetd(ADMN)

The **telnetd**(ADMN) manual page incorrectly includes a description for a **/-D** option. This option is not available.

slattach(ADMN)

The **slattach**(ADMN) manual page should also include the following available option under the heading "Options":

{+ | -} **f**   enable/disable hardware flow control

Remote line printing

In Chapter 7, "Remote line printing" of the *SCO TCP/IP User's and Administrator's Guide*, there is a section named "Using RLP" which has a subsection named "SCO clients". In this subsection the command **lpstat** is referenced as a command to use to acquire information about remote printers. The correct command to use is **rlpstat**. On the page referenced here, replace **lpstat** wherever it appears with **rlpstat**.

PPP (Point-to-Point Protocol) administration

The following information pertaining to the updated PPP implementation belongs in the *SCO TCP/IP User's and Administrator's Guide* in Chapter 8, "Administering serial line communications" under the section "Administering PPP."

The section "Preparing to configure PPP" includes a bullet list of information you need to know when configuring PPP. This list should also include the following:

- remote system's ID and password if you want to use the Password Authentication Protocol. For more information see the online reference manual page *pppauth*(SFF).

The following information should be changed in the "More PPP information" section:

- Add the following to the table of manual pages.

  pppauth(SFF) — PPP authentication database

- Replace the last paragraph of the chapter with the following: For more information about how PPP is designed and implemented, refer to RFC 1331, RFC 1332 and the Internet Draft "The PPP Authentication Protocols" (the Draft is by B. Lloyd and W. Simpson). See the online **ppp**(ADMP) manual page for a description of what is not implemented from these documents. These documents are available from the Network Information Center in Chantilly, Virginia.

See the next section in these *Release and Installation Notes*, "Domain and subdomain registration", for complete information on contacting the Network Information Center.

The numbered procedure in the section "Configuring PPP with netconfig" should have the following additional steps:

12a. At the prompt `Enter local host name or enter q to quit [ID]:`, enter the name of this system; in this example, **emerald**. This name will be used in the "peer_id" field of a sent PAP Authenticate_Request message.

12b. At the prompt `Enter local host password for PPP authentication or enter q to quit [PASSWD]:`, enter the password that the peer also knows. The password will be used in the "password" field of a sent PAP Authenticate_Request message.

14a. When **netconfig** prompts you to add or remove TCP connections, enter **q** if you want to maintain the TCP connections currently configured. If you want to change the number of TCP connections, enter **y**.

The section "Adding PPP information to configuration files" requires one correction to existing information and requires new information on two additional configuration files.

The correction is to the last line of the paragraph describing the */usr/lib/uucp/Systems* file example on page 69. The existing line states that "Both lines have **testing** as their login keyword." The word "testing" should be "nppp".

Following is new information about the configuration files */etc/ppphosts* and */etc/pppauth*. Additional information about these files is found in the online manual pages *ppphosts*(SFF) and *pppauth*(SFF). Some entries in these files, which are explained below, are made automatically at configuration time. All other additions and changes to these files must be made manually. After editing these files, make sure the permissions modes for both files are "644" to maintain system security.

/etc/ppphosts:
The online manual page *ppphosts*(SFF) provides an explanation of the file syntax along with examples illustrating the importance of having entries in this file correspond with entries in the */etc/hosts, /etc/lib/uucp/Systems,* and */etc/lib/uucp/Devices* files.

When the command **netconfig** is run to configure PPP, the following entry is made automatically in /etc/ppphosts:

```
*nppp - - idle=3 conf=5
```

This entry, or a similar one beginning with " * ", must be present for remote hosts to request a connection with the local host. A remote host uses the value in the first field as a login name. **netconfig** also creates a corresponding user login named "nppp". Unlike a typical user login, however, **netconfig** assigns */usr/lib/ppp* as the home directory and */usr/lib/ppp/ppp* as the shell. **netconfig** prompts for a login password.

Additional login entries may be added to this file manually. Any additional entries must also have corresponding user logins created with the home directory and shell set to the files described above. The local host uses parameters in these entries for incoming connection negotiation and control.

/etc/pppauth:

```
*emerald mysecret
laiout.i88.isc.com peersecret
```

This file is the repository for password information used in PPP password authentication (separate from login password authentication) as specified in the PPP Authentication Protocol (PAP). Password authentication only occurs if requested by one or both sides of a PPP connection during negotiation. Password authentication is requested by the local host if the "pap" link option is present in the */etc/ppphosts* file entry governing the connection being negotiated.

When the local host requests password authentication, the remote host sends an Authenticate_Request message containing a peer-id value and a password value. For the local host to permit a PPP connection to occur, these values must exist as a pair in the */etc/pppauth* file of the local host. The second entry above shows peer_id and password values for a remote (also called "peer") host.

If a remote host requests password authentication, PPP on the local host retrieves the peer_id and password values from the entry beginning with " * " and sends that to the remote host within an Authenticate_Request message. If these values do not match those expected by the remote host, the connection is denied by the remote host. The first entry above shows a typical local host name (indicated by " * " at the beginning of the entry) and password.

The following new section should be added to the "Administering PPP" section:

Interoperability with other PPP implementations

The following paragraphs provide configuration details for establishing PPP connections between the current implementation of PPP (in SCO TCP/IP Release 1.2.1) and a system running the PPP version in SCO TCP/IP Release 1.2.0 or PPP versions from other vendors.

For connections with SCO TCP/IP Release 1.2.0 PPP:

On the system running SCO TCP/IP 1.2.1 PPP, include the following options in the */etc/ppphosts* file:

- Link option "accomp".
- IP options "rfc1172addr", "ipaddr", and "VJ".
- Other option "old".

For connections with MorningStar PPP 1.3 for SCO:

On the system running MorningStar PPP, use the "nolqm" option on the **pppd** command line. This is because the SCO TCP/IP 1.2.1 PPP implementation does not support link quality control. Also, do not use password authentication on both sides at the same time since MorningStar PPP 1.3 only supports one side password authentication.

For dial-up connections from FTP PC/TCP version 2.05:

On the system running SCO TCP/IP 1.2.1 PPP, include the options "ipaddr", "accomp", and "rfc1172addr" in the */etc/ppphosts* file. In the same file, also set *mru*=1500.

On the system running FTP PC/TCP version 2.05, set *mru*=1500, set *mtu*=1500, and turn on the "address/control field compression" flag.

For connections with a Telebit NetBlazer 1.5:

Connecting to a NetBlazer from a UNIX System:

1. Configure the NetBlazer to receive the call from the UNIX system:

   The NetBlazer may be set up as if the UNIX machine were just another remote dynamic-interface NetBlazer, using the PPP encapsulation protocol in packet mode.

   Configure the NetBlazer according to the Dynamic Interface Configuration procedure found in the NetBlazer Installation Guides, noting particularly the discussion of the **ipdial** command. When you are prompted for `Name of remote(other) system:`, enter the name of the UNIX system. Use

"PPP" rather than " SLIP". Set ***mtu***=1500. Make a note of the dial-in user password you enter for use by the UNIX system; you will need to include this password in the UNIX system's */usr/lib/uucp/System* entry that describes how to dial into this NetBlazer (see next step).

2. Configure the UNIX system to call the NetBlazer:

   In your system's */usr/lib/uucp/Systems* file, set the username provided by the login portion of the chat script to the UNIX system name. Make sure you set the password to the same one you entered for the dial-in user on the NetBlazer (see step 1, above). You must turn off the ipaddr option in the */etc/ppphosts* file because NetBlazer rejects ip-address negotiation.

Connecting to a UNIX system from a NetBlazer:

1. Configure the UNIX system to receive the call from NetBlazer:

   Turn off the ipaddr option in the */etc/ppphosts* file because NetBlazer 1.5 rejects IP-address negotiation.

2. Configure the NetBlazer to call the UNIX system:

   Add the following entry to the NetBlazer's *CHAT.TXT* file:

```
# for connecting to a SCO UNIX machine.
:unix
s1,20,30,2,
e2,20,30,3,in:
e3,3,4,4,$6%~&
u4,5,100,5
s5,5,100,6,
e6,20,30,7,word:
e7,3,8,8,$6%~&
p8,5,100,9
s9,5,100,99,

s30,20,30,31,
e31,20,30,32,in:
e32,3,33,33,$6%~&
u33,5,100,34
s34,5,100,35,
e35,20,100,36,word:
e36,3,37,37,$6%~&
p37,5,100,38
s38,5,100,99,

r99,0
```

   Reboot the NetBlazer to effect any changes to the *CHAT.TXT* file. When you configure the NetBlazer's dynamic interface to call the UNIX system (as described in "Configure the NetBlazer to receive the call from the UNIX system", above), specify `unix` for the chat script instead of the default `ics`.

Domain and subdomain registration

To register a top-level domain and subdomain name, call or write to the NIC (Network Information Center) standards organization at the following address:

Network Solutions
AttN: InterNIC Registration Services
505 Huntmar Park Drive
Herndon, VA 22070
USA

Help desk telephone number is:

1-703-742-4777

Help desk hours of operation:

7:00 am to 7:00 pm Eastern Time

Email address:

hostmaster@internic.net	Host, domain, network changes and updates
action@internic.net	Computer operations
mailserv@rs.internic.net	Automatic mail service

Network address:

198.41.0.5 (RS.INTERNIC.NET)

SMUX (SNMP Multiplexing) protocol information

The following information pertaining to the new SMUX protocol feature belongs in Chapter 11, "Configuring and using SNMP," of the *SCO TCP/IP User's and Administrator's Guide.*

SMUX protocol

On a typical system, an SNMP agent gathers the information pertinent to objects in the Internet standard MIB by reading variables from the kernel or stable storage (i.e., files). Unfortunately, user-processes are often employed to perform many network services and the information the processes maintain cannot be easily accessed by the SNMP agent. SNMP cannot be used to manage the information maintained by user-processes unless the SNMP agent can communicate with the user-processes.

The SMUX (SNMP multiplexing) protocol defines a mechanism by which multiple user-processes can communicate with the local SNMP agent. This enables the SNMP agent to manage objects whose data does not reside in areas directly accessible to it. This allows the set of objects managed by the local SNMP agent to vary dynamically.

For a user-process to be permitted to export a portion of the MIB (that is, establish a relationship with the local SNMP agent to manage objects for which the user-process maintains data), the user-process must be listed in the SNMP agent's */etc/snmpd.peers* file; such a process is called an "SMUX peer." When a user-process wishes to export a subtree of objects, it establishes a TCP connection with the local SNMP agent, registers the sub-tree of objects it wishes to export, and later on fields SNMP agent queries for objects in this sub-tree. The SNMP agent uses the TCP port 199 for listening to incoming requests from potential SMUX peers.

For more information on the SMUX protocol refer to RFC 1227. Also, see the online reference manual pages **snmpd**(ADMN) and **snmpd.peers**(SFF).

SCO®
OPEN SYSTEMS SOFTWARE

Please help us to write computer manuals that meet your needs by completing this form. Please post the completed form to the Publications Manager nearest you: The Santa Cruz Operation, Ltd., Croxley Centre, Hatters Lane, Watford WD1 8YN, United Kingdom; The Santa Cruz Operation, Inc., 400 Encinal Street, P.O. Box 1900, Santa Cruz, California 95061, USA or SCO Canada, Inc., 130 Bloor Street West, 10th Floor, Toronto, Ontario, Canada M5S 1N5.

Volume title: ____________________

(Copy this from the title page of the manual)

Product: ____________________

(for example, SCO UNIX System V Release 3.2 Operating System Version 4.0)

How long have you used this product?

□ Less than one month □ Less than six months □ Less than one year

□ 1 to 2 years □ More than 2 years

How much have you read of this manual?

□ Entire manual □ Specific chapters □ Used only for reference

	Agree				*Disagree*
The software was fully and accurately described	□	□	□	□	□
The manual was well organized	□	□	□	□	□
The writing was at an appropriate technical level (neither too complicated nor too simple)	□	□	□	□	□
It was easy to find the information I was looking for	□	□	□	□	□
Examples were clear and easy to follow	□	□	□	□	□
Illustrations added to my understanding of the software	□	□	□	□	□
I liked the page design of the manual	□	□	□	□	□

If you have specific comments or if you have found specific inaccuracies, please report these on the back of this form or on a separate sheet of paper. In the case of inaccuracies, please list the relevant page number.

May we contact you further about how to improve SCO documentation? If so, please supply the following details:

Name ____________________ *Position* ____________________

Company ____________________

Address ____________________

City & Post/Zip Code ____________________

Country ____________________

Telephone ____________________ *Facsimile* ____________________

27 August 1993