Tru64 UNIX

Technical Overview

Part Number: AA-RH8XA-TE

July 1999

Product Version: Tru64 UNIX Version 5.0 or higher

This document describes the components of the Compaq Tru64 UNIX (formerly DIGITAL UNIX) operating system.

Compaq Computer Corporation
Houston, Texas

Contents

7 Security

8 Internationalization

Index

Figures

Tables

About This Manual

This document provides a brief technical overview of the Compaq Tru64™ UNIX® (formerly DIGITAL UNIX) operating system and its components.

This book is not intended to supersede the *Software Product Description* (SPD), which is the definitive legal document describing the functionality in Tru64 UNIX that Compaq supports.

Audience

This manual is for anyone who is interested in the components that make up Tru64 UNIX.

Organization

This document contains the following chapters and appendixes:

Chapter 1	Introduces Tru64 UNIX and discusses recent enhancements, available licensing, and available optional software.
Chapter 2	Discusses the various features for system management.
Chapter 3	Describes the available networking protocols and applications.
Chapter 4	Describes the various file systems supported by Tru64 UNIX.
Chapter 5	Discusses issues regarding the kernel; these issues include system attributes and parameters, symmetric multiprocessing, and virtual memory.
Chapter 6	Highlights the major features provided by the Tru64 UNIX development environment.
Chapter 7	Discusses how Tru64 UNIX addresses the requirements of the C2 evaluation class and additional security features.
Chapter 8	Discusses issues concerning Internationalization.

Related Documents

- The *Software Product Description* (SPD) , # 70.70.*xx*, where *xx* indicates a given release).
- This Technical Overview refers to various Request for Comments (RFC). RFCs define the Internet protocols and policies; they are available from numerous sites including the following:
  http://www.ietf.cnri.reston.va.us/home.html
  http://www.cis.ohio-state.edu/hypertext/information/rfc.html
- ActiveAnswers offers a comprehensive solution web-based program, which includes several tools that can be used for planning, deploying, operating, and maintaining key applications. These tools assist applications in many areas including, but not limited to, management, database and business applications, high availability, and Internet solutions.

  The main purpose of ActiveAnswers is to accelerate the time to solution while also positioning Compaq as a key enterprise solution provider. Partnerships have been made with key Independent Software Vendors (ISVs) such as Microsoft, SAP, Novell, Oracle, and Genesys. Web-based solutions for customers and system integrators have also been developed. Please visit the ActiveAnswers web page at
  http://vcmproapp02.compaq.com/ActiveAnswers/global/en/home
- The Tru64 UNIX documentation set, which is available on the Tru64 UNIX CD-ROM in both PDF and HTML format.

  The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the books to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Compaq.) The following list describes this convention:

G	Books for general users
S	Books for system and network administrators
P	Books for programmers
D	Books for device driver writers
R	Books for reference page users

  Some books in the documentation help meet the needs of several audiences. For example, the information in some system books is also used by programmers. Keep this in mind when searching for information on specific topics.

  The *Documentation Overview* provides information on all of the books in the Tru64 UNIX documentation set.

Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

- Mail:

  Compaq Computer Corporation
  UBPG Publications Manager
  ZKO3-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

  A Reader's Comment form is located in the back of each printed manual. The form is postage paid if you mail it in the United States.

Please include the following information along with your comments:

- The full title of the book and the order number. (The order number is printed on the title page of this book and on its back cover.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

Conventions

This guide uses the following conventions:

file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
`cat`(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.
MB/s	This symbol indicates megabytes per second.
Mb/s	This symbol indicates megabits per second.

1

Introduction to Tru64 UNIX

This chapter introduces Tru64 UNIX, provides a description of the operating system, describes recent enhancements, and available optional software.

1.1 Introduction

Compaq Tru64 UNIX operating system Version 5.0 delivers many new and significant features that bring Tru64 UNIX to the highest level of performance, scalability, and availability, yet make it simple to manage and operate. Tru64 UNIX Version 5.0 continues to offer warranted Year 2000 readiness (the first Year 2000-ready release was V4.0D in January 1998). For users seeking to expand their information technology capabilities while increasing performance and maintaining the highest level of availability, Compaq offers the best industry UNIX solution.

Compaq Tru64 UNIX is the most time-tested 64-bit UNIX operating system you can buy. It brings new strengths, with features that:

- Dramatically simplify system management by offering a choice of management interfaces
- Substantially reduce the complexity of installation, setup, and management
- Reduce the total cost of ownership by offering familiar interfaces, tools with a common look and feel, and automation, thereby minimizing the need for training
- Grow to multiple terabyte configurations, giving users the flexibility to satisfy their growing business needs
- Increase performance in file system, storage management, and system networking
- Deliver very high integration between UNIX and Windows NT
- Fill users' highest availability needs with the addition of Compaq TruCluster Server Software Version 5.0

1.1.1 Product History

Tru64 UNIX is the Compaq implementation of the Open Software Foundation Version 1.0 and Version 1.2 technology, and the Motif Version 1.2.5 graphical user interface and programming environment. In addition, Tru64 UNIX supports the full features of the X Window System, Version 11, Release 6.3 (X11R6.3) from MIT.

The Tru64 UNIX operating system is a multiuser/multitasking, 64-bit, advanced kernel architecture based on Carnegie Mellon University's Mach Version 2.5 kernel design with components from Berkeley Software Distribution (BSD) Versions 4.3 and 4.4, UNIX System Laboratories System V Release 4.0, other software sources, the public domain, and from Compaq Computer Corporation.

Previous versions of Tru64 UNIX were known as DIGITAL UNIX.

1.1.2 Product Description

Tru64 UNIX is supported on all Alpha servers and workstations with 64 MB of memory.

The operating system incorporates several performance enhancements either developed or extended by Compaq, including Wired Memory, Virtual Memory, and Unified Buffer Cache; UFS file block clustering and cached writes over NFS; IP Multicasting, path MTU discovery, and optimized TCP/IP; and quick-started shared libraries.

In addition, Tru64 UNIX provides a clear and concise system administration environment (including graphics, Web-based, and character-cell) to:

- Simplify system administration tasks
- Enable an update installation that does not overwrite system files and a new and improved full installation that allows you to get up and running almost immediately while files are being copied from the CD-ROM onto your system disk
- Supports loadable drivers and other kernel subsystems, including loadable boot-path support for third-party disks and graphics cards
- Provide support for dynamic system configuration and dynamic system recognition of disks and tapes

Tru64 UNIX supports the Common Desktop Environment (CDE) as the default user interface. CDE provides a uniform graphical user interface, portable across multiple platforms, to facilitate common end-user and system administration tasks.

Tru64 UNIX provides real-time support and symmetrical multiprocessing (SMP), dataless servers and clients, and numerous features intended to assist application programmers in developing applications that use shared libraries, threads, and memory mapped files. It is fully compliant to the Single UNIX Specification, to the X/Open UNIX brand, to POSIX 1003.1B (Real-time) and to POSIX 1003.1C (with DECthreads).

1.1.3 Standards

To ensure a high level of compatibility with the Compaq ULTRIX operating system, the Tru64 UNIX operating system is compatible with the Berkeley 4.3 and System V programming interfaces and, by complying with the System V Interface Definition (SVID3 Base and Kernel Extensions), Tru64 UNIX supports System V applications as well.

Part of the charter of The Open Group (formerly The Open Software Foundation) was to provide an interface for developing portable applications that would run on a variety of hardware platforms. Tru64 UNIX is compliant with the OSF Application Environment Specification (AES) that specifies the interface to support these portable applications. In addition, the Tru64 UNIX operating system complies with standards and industry specifications, including FIPS, POSIX, X/Open, XTI, and AT&T System V Interface Definition (SVID).

For a complete list of the standards that Tru64 UNIX supports, see the Software Product Description (SPD).

1.2 Product Features and Enhancements

Tru64 UNIX Version 5.0 offers an array of significant new features and enhancements to the operating system to support Enterprise applications, high performance technical computing, business intelligence, and Internet communications. The following sections discuss some of the most important features; Table 1–1 gives a one to one correlation of the benefits of these features.

Table 1–1: Features and Benefits

Features	Benefits
More Graphical User Interfaces	Easy facilitation of tasks including Storage Administrator, Bootable Tape, Advanced File System, Enhanced Security Audit Configuration and Setup, and DOS Tools.
Device Naming	Up to 256 target and LUN addresses; devices retain their name even when moved to a different SCSI address
Advanced File System	High performance, online backups and system administration, disk spanning.

Table 1–1: Features and Benefits (cont.)

Features	Benefits
SysMan Menu	Single consistent presentation format for administrative utilities. Integration of most system administration utilities, hierarchical task-oriented menu for launching utilities, accessible under several different user environments.
SysMan Station	Single consistent presentation format, graphical representation of a system (or cluster) monitoring status from the CPU down to the level of individual subcomponent.
Quick Setup utility	Streamlines the configuration of client systems. Also can be used to perform initial setup for servers, then the System Setup utility (also known as the Checklist) can be used to complete the configuration.
System Cloning	Speeds up the installation of multiple systems by duplicating the file system layout, file system type, and software subset selections from one installed system onto others.
Insight Manager	Allows monitoring of Compaq systems (including Tru64 UNIX systems, Windows NT workstations, PCs, and laptops) in a network. Some systems can be administered remotely.
NetRAIN support	Improves reliability of transmitted data by allowing failover from one controller to another when a fault is detected.
Multiple access paths to storage	High availability through more than one access point to a storage item. When one path is blocked because the network is down, or a cable is pulled, for example, another path takes on the workload. Applications continue without disrupting users who need the stored information.
Event Management	Event Manager provides a centralized means of gathering, distributing, storing, and retrieving event information, regardless how the events are posted. Now all system event information (hardware and software) is more accessible and further enhances system up-time.
Direct file system I/O access	This feature enables read and write requests on a file; that is, the ability to be executed to and from disk storage through direct memory access, bypassing AdvFS caching. This is a tremendous performance improvement on the I/O process for applications, especially database applications.

Table 1–1: Features and Benefits (cont.)

Features	Benefits
File system Backup API	Consistent application interface that gives all third party products and customer-developed products an easy way to guarantee that backups and restores are effective and efficient.
Enhanced dynamic tuning	Most kernel parameters are now able to be tuned without bringing down the system, enabling you to maximize your performance on the fly.

1.2.1 Scalability Enhancements

Tru64 UNIX includes the following scalability enhancements:

- More efficient scaling with more CPUs. Enhanced scaling allows for the addition of more CPUs in SMP systems. Additionally, several design improvements have been made to allow increased CPU scaling with future systems.
- Very Large Memory (VLM). Improved memory utilization to 28 GB, which provides more efficient use of memory in VLM configurations.
- File and Storage Limits. Supports 4 terabytes or larger file and storage systems.
- Performance benchmarks. Provides improvements in the following benchmarks: AIM VII, LADDIS, TPC-C, TPC-D, and SPECweb.
- Enhanced device support:
  - SCSI-3 standard support allows up to 256 target and LUN addresses per SCSI bus and supports dual pathing (alternate path to a device).
  - New device naming provides a flexible framework that allows more than 256 device names per SCSI bus. The new device naming supports SCSI-3 and FibreChannel. It includes the ability to convert existing device names in an existing configuration to new device names automatically, while retaining support for old device names.

    The advantages for the new device naming include the ability to retain its name, even if it moves to a different SCSI address and support for multipathing to a disk where the disk is accessible through a different SCSI controller.
  - Support for FibreChannel switched connections and multiple concurrent paths (64) with adaptive load balancing has been added. Both SCSI-2 and SCSI-3 commands are supported.
- Dynamic Tuning. Most tuning can now be done at run time without rebooting the system.

1.2.2 Enhancements to the Installation Process

The following enhancements have been made to the Full and Update Installation processes:

- Your system must have a minimum of 64 MB of memory to install and run this version of the operating system.
- If you plan to install all file systems and swap space on a single disk, the disk must be 1 GB or larger.
- If you decide to install optional software either during a Full Installation or after the system is installed, the `setld` utility automatically checks software dependencies. Dependency checking ensures that software required for the proper operation of the optional software is also installed. If such a relationship is detected, the dependent software is installed automatically and you are notified accordingly.

Changes to the Update Installation process include:

- Your system must be running Version 4.0D or 4.0F of the operating system in order to update to Version 5.0. The *Installation Guide* shows the successive update paths to reach Version 5.0 if your system is running a version of the operating system other than Version 4.0D or 4.0F.
- A graphical user interface has been added to the Update Installation process. In previous releases, the Update Installation only provided a text-based interface.
- Updating Worldwide Language Support (WLS) software is performed automatically during an Update Installation of the base operating system. It is no longer necessary to remove WLS software before the operating system is updated or update WLS software as a separate task.
- The Update Installation can be invoked with the optional `-u` flag to run the Update Installation in **unattended** mode. Unattended means that barring any problems with the update, there is no user interaction required. The only exception to this is the switching of CD-ROMs if WLS software is being updated. The `-u` flag builds a kernel with all kernel components and does not provide the chance to archive obsolete files.
- An **analysis phase** has been added to the beginning of the Update Installation process. Your system is analyzed for the following:
  - Layered products that prevent the Update from continuing
  - Layered products that should be reinstalled after the Update
  - Fatal and non-fatal file system type conflicts
  - Available disk space

If layered product or non-fatal file type conflicts are discovered, you can resolve them directly from the Update user interface; there is no need to exit the Update, resolve the conflict, and restart the Update. If your system does not have enough available disk space for new software and room for Update processing, disk space recovery options are available directly from the Update Installation as well.

Major changes to the Full Installation include:

- Both the text-based and graphical user interfaces have a new task oriented design, which steps you through each installation task and lets you go backward and forward at any time to change your answers.
- By default, the Full Installation process determines the file system layout based on your software selections. You do not need to calculate in advance the size of the file systems, nor do you need to repartition your disks in advance of the installation process to ensure a successful installation.
- Partition `a` of the disk you choose to hold the root file system must be at least 128 MB in size.
- If you are installing the operating system onto a single disk, the recommended partition table creates a 745 MB `g` partition and a 128 MB `b` partition for swap space.
- The Advanced File System (AdvFS) is now the default file system type. You still have the option to use the UNIX File System (UFS).
- The Logical Storage Manager (LSM) can be installed and configured during a Full Installation, eliminating the need to install and configure LSM as a separate task.
- Worldwide Language Support (WLS) software can be installed during a Full Installation, eliminating the need to install WLS software as a separate task.
- User supplied files can be invoked during a Full Installation to further customize the system beyond what is possible during a regular Full Installation. User-supplied files can contain scripts, executables, or programs. A new invocation point and file have been added to this feature. The Full Installation now also searches for a file named `postreboot`, which is searched for and executed after the system reboots, but before the tailored kernel build and configuration phase.

1.2.3 System Management Enhancements

Tru64 UNIX provides several new applications for the configuration and daily administration of a system. Tru64 UNIX System Management consists of a suite of tools for configuration and for daily management of a

Tru64 UNIX system. The following sections provide a brief overview; for more information on these applications, see the corresponding online help for the application and the *Network Administration* and *System Administration* guides.

1.2.3.1 System Management Menu

The System Management Menu (SysMan Menu) provides a framework for organizing various system management tasks. Each task represents a small application that is launched from the SysMan Menu.

The SysMan menu can be run in a CDE, HTML, or ASCII text environment. Thus, all the tasks on this menu can be performed from an X11 capable display, a personal computer running Windows 95, Windows 98, or Windows NT, or from a character cell terminal.

1.2.3.2 System Management Station

The System Management Station (SysMan Station) provides a graphic representation of the Tru64 UNIX system and enables you to manage it remotely from a personal computer. This Java tool is fully integrated with TruCluster Server software Version 5.0; it allows for remote management from anywhere and from any computer: PC clients, UNIX workstations, and any Alpha system.

1.2.3.3 Quick Setup

Quick Setup guides you through the essential and most commonly performed configuration steps. It provides a fast, user-friendly way to set up your system with the basic system configuration. The resulting system can be used “as is” or you can augment it with settings accessible in the full-featured configuration applications. On a character cell system, the Quick Setup application is available from the System Setup menu (`/usr/sbin/setup`).

1.2.3.4 Division of Privileges

All privileged system management applications launched via the SysMan Menu, the SysMan Station, the desktop's Application Manager, or the Custom Setup checklist/menu use Division of Privileges (DoP). The DoP user interface, `sysman dopconfig`, enables administrators to grant users or groups access to execute these privileged programs without the root password.

1.2.3.5 Network Setup Wizard

The Network Setup Wizard leads you through the various applications to add a system to a network. The wizard steps you through the applications in a recommended order and provides information to help you determine which applications are applicable to each situation.

1.2.3.6 ATM configuration

There is a new SysMan application for configuring the ATM subsystem. This application supports configuring Classical IP over Permanent Virtual Circuits (PVCs) and IP Switching.

1.2.3.7 Point-to-Point Configuration

The Point-to-Point (PPP) Option File configuration application offers a simplified method of creating and deleting PPP option files and for selecting options and their values. The PPP Authentication File configuration allows you to add, delete, or modify an entry in the `pap-secrets` or `chap-secrets` authentication files. These applications eliminate the need to create or modify these files manually.

1.2.3.8 Network Time Protocol Configuration

The Network Time Protocol (NTP) configuration application enables you to set up an NTP client or peer (a local NTP server that can synchronize the time with another server or be synchronized by it). It also enables you to check the status of the NTP daemon and to start, restart, and stop the daemon.

1.2.3.9 Domain Name Service Configuration

The Domain Name Service (DNS) configuration application has a new interface, which presents you with a list of tasks relating to DNS/BIND. This application replaces the `bindsetup` and `bindconfig` applications.

1.2.3.10 Network File System Configuration

The Network File System (NFS) configuration application presents you with a list of tasks relating to NFS setup.

1.2.3.11 System Cloning

Tru64 UNIX includes the ability to duplicate the file system layout, file system type, and software subset selections from identical systems that are

already installed. Performing a cloned installation considerably speeds up the installation procedure.

1.2.3.12 Service Tools

Tru64 UNIX provides graphical presentation of the following commands:

- `iostat` (I/O statistics)
- `netstat` (network statistics)
- `systemmessages` (system messages)
- `vmstat` (virtual memory statistics)
- `who`

Tru64 UNIX also supports the following:

- DECevent, which provides error reporting and binary-to-text translation capabilities. DECevent provides system directed diagnostic capability for various platforms.
- Compaq Analyze is a rules-based hardware fault management diagnostic tools that provides error event analysis and translation. Its multievent correlation analysis provides the capability to analyze events stored in the system's event log file. The graphical user interface enables users to set program and configuration parameters and to review event information.
- UniCensus is a tool for collecting and archiving system configuration information to produce an HTML report showing system configuration information, revision levels, storage subsystem configuration, and other information. UniCensus can be configured to run on a system reboot.
- The `sys_check` command outputs the system configuration, hardware and software, of the running system and displays the information in HTML format. The `sys_check` command also performs a basic analysis of operating system parameters and attributes and provides warnings if it detects problems.

1.2.3.13 Bootable Tape

The Bootable Tape application is a graphical user interface that gives you the ability to create and recover a disk image from a system.

1.2.4 File Systems and Storage

The following sections describe new features for the AdvFS file system, which is now the default file system for the operating system.

1.2.4.1 Fileset Creation with var Area in /usr

Selecting the /usr option for the var area now creates both a usr fileset and a var fileset within the usr domain when AdvFS is selected for the usr file system. You can continue to create separate filesets, but take advantage of AdvFS's ability to share disk resources between filesets. Previously, this option created a single usr fileset within the usr domain that contained both usr and var information.

This change affects users who back up AdvFS filesets with the vdump command. Previously, if your system was configured with var in the usr fileset, you needed only to back up the usr fileset. Now that this option creates a separate fileset for the var area, you must issue an additional vdump command against the var fileset or the information in the var fileset will not be backed up.

1.2.4.2 On-Disk Format for AdvFS Domains

This release provides a dramatically improved AdvFS on-disk file system structure that increases performance on AdvFS file systems that have a large number of files.

For detailed information, see the *AdvFS Administration* guide.

1.2.4.3 Storage Management

NetWorker™ is an application that provides automated backup and recovery of files on a single local system. In addition to the backup product Networker, the Logical Storage Manager uses RAID technology to provide highly available and customizable storage solutions for mission-critical environments. It is an integrated, host-based disk management tool that enables administrators to configure storage, protecting against data loss and improving disk I/O performance.

1.2.5 Networking Features

Networking enhancements include the following:

- Fast Ethernet support (IEEE 802.3 100Base-TX in full and half duplex).
- Gigabit Ethernet support (IEEE 802.3z Gigabit Ethernet Standard, IEEE 802.3x Pause Frame Flow control (X-on/X-off), both symmetric and asymmetric, and Jumbo frame compatible.
- NetRAIN support is provided for Ethernet, FDDI, and ATM controllers; it allows for the failover of communications from one controller to another in the event a fault is detected in the communications path.

- ATM enhancements, including: a new SysMan configuration application. You no longer need to configure Classical IP over PVCs and IP Switching through the `/etc/atm.conf` file.

  This file is still supported for starting user-written or third-party ATM components that are not supported by the ATM configuration application. See the *Software Product Description* for further information. The `atm.conf` file is now automatically invoked during system booting, immediately after the ATM configuration has been performed.
- Support for RFCs 1901-1908 (SNMPv2C) in the extensible SNMP agent, subagent developer's tools, and SNMP-related commands.
- Support for RFC 2089 (Mapping SNMPv2 to SNMPv1) in the bilingual extensible SNMP agent.
- Support for RFC 2257 (Agent Extensibility) in the extensible SNMP agent and subgent developer's tools.
- PPP has been updated to Version 2.3.1 and a new configuration application has been added.
- Resource Reservation Protocol (RSVP) (RFC 1633) has been added for Ethernet and FDDI, which provides Quality of Service (QoS) for real-time traffic.
- The Domain Name Service (DNS) has been upgraded from BIND Version 4.9.3 to Version 8.1.1, including dynamic updates to DNS.
- New TCP/IP features, including Path MTU over UDP and enhanced `rlogin` and `telnet` scaling.
- Support for a single system to have multiple active network adapters in the same subnet.
- Support for a single system to have multiple network adapters in the same subnet that are configured to provide failover in case of an adapter failure.

1.2.6 Monitoring and Tuning

Tru64 UNIX offers several monitors:

- Kernel Tuner, which lets you display and change parameters of the kernel subsystem,
- Class Scheduler, which allows the operating system manager to prioritize tasks.
- Process Tuner is an application to display, monitor, and manage the system processes. A number of sort and filter options are provided to help manage the display of the information.

- Performance Manager is a real-time application that provides tools for detecting and correcting performance problems
- Environmental monitoring minds the thermal, fan, and redundant power supply state of Alpha Server systems with prerequisite hardware sensor support.

1.2.7 UNIX/NT Interoperability

Most corporations today own a mix of technology that has been purchased over a number of years. Some installations are based largely on UNIX and are just beginning to include Microsoft's Windows NT. Others are Windows-based and are just beginning to incorporate UNIX. It is important to understand the unique needs of each of these types of organizations for planning for Windows NT integration. Industry analysts estimate that ninety—five percent of all large organizations will be using both UNIX and Windows NT operating systems by the year 2000. The market-wide customer challenge – integrating UNIX and Windows NT to improve customers' competitive advantage – is the focus of the Compaq Windows-Friendly Tru64 UNIX Program.

Compaq's integration solution begins with customers' current assets – software, data, and skills. Building on current technology and user knowledge base, Compaq speeds integration between UNIX and Windows NT, thereby increasing their combined contribution to the enterprise's productivity.

Compaq provides a number of capabilities and products with Tru64 UNIX to simplifying the development, deployment and management of solutions in a mixed UNIX/Windows NT environment. These capabilities help solve customer integration problems across a number of functional areas. Integration between Windows NT and UNIX is solved through the use of Advanced Server for UNIX, Compaq Insight Manager, COM for Tru64 UNIX and driver support (ODBC and JDBC).

1.2.8 Advanced Printing Software

The Advanced Printing Software (APS) is a new printing system for Tru64 UNIX developed in collaboration with Xerox, and based on the PrintXchange technology from Xerox. APS is a distributed client/server printing system for workgroup and enterprise environments. Based on the ISO 10175 Document Printing Application and POSIX 1384.7, APS uses inbound and outbound gateways to move print jobs to or from the `lpr`/`lpd` print subsystems.

1.2.9 Documentation

The documentation set has been updated to make it more comprehensive and easier to use. Some of the major improvements are:

- The *Installation Guide* has been divided into two volumes to address the needs of different users. The *Installation Guide* has been reduced in scope to make it easier for the novice user. The new *Installation Guide — Advanced Topics* guide provides expert installers with the information required to perform custom installations.
- The *System Administration* guide has been updated to make it more task-oriented.
- The online help for SysMan and its associated applications has been updated to make it more complete and robust and task-oriented to help you find the information you need to complete a given goal. Online help has been developed to support new applications.
- The complete set of base operating system reference pages is now included in HTML form on the Documentation CD-ROM, with access through the standard HTML Documentation Library. The reference pages are indexed both alphabetically and by section. Reference pages for layered products are not included on the Documentation CD-ROM; if they are installed, those reference pages can be viewed with the `man` command, the `xman` command, or the webman viewer.

  With the addition of the HTML reference pages, the webman viewer has been changed so that it displays only reference pages that actually are installed on the system. The redundant set of compressed reference pages has been removed from the Documentation CD-ROM.

  The AltaVista search tool is now bundled in to help you search for the topic you want. The AltaVista software finds the appropriate references.

1.2.10 Y2K Readiness

Tru64 UNIX is Year 2000 Ready. Compaq defines this as products capable of accurately processing, providing, or receiving date data from, into, and between the twentieth and the twenty-first centuries, and the years 1999 and 2000, including leap year calculations, when used in accordance with the associated product documentation, and provided that all hardware, firmware, and software used in combination with such products properly exchange accurate date data with the products. For additional information visit the DIGITAL Brand area on Compaq's Year 2000 web site located at: `http://www.compaq.com/year2000`.

The following testing process and methods were used to ensure that Tru64 UNIX is Year 2000 Ready:

- Code Inspection

  All source code modules used to build this product were inventoried and inspected to ensure correct date handling for date data beyond the year 2000

- System Date Handling

  This product was tested to ensure that the system properly handles future time including, but not limited to, the following dates:

  – December 31, 1999 to January 1, 2000 rollover

  – February 28, 2000 and February 29, 2000

  – March 1, 2000

  – January 1, 2001

- Regression Testing

  This product was tested using a comprehensive suite of regression tests for functional, performance, and standards compliance with system time set to future dates including dates in and beyond the year 2000.

To ensure that this product interoperates properly with other hardware and software, the following testing process/methods were used:

- Year 2000 readiness was tested using supported hardware and firmware.
- This product has been tested for Year 2000 readiness while operating within a computer network of other systems.
- The Associated Products (shipped with the Tru64 UNIX media kit) have been tested in conjunction with the operating system for Year 2000 readiness.

1.3 Performance

Compaq has improved the performance of the operating system and many applications. Examples include the following:

- E-business

  Substantial improvements in the kernel and in the networking support now enable faster and more reliable Internet.

- Kernel tuning

  The default values for the `maxusers`, `vm-page-free-target`, `vm-mapentries`, and `vm-vpagemax` kernel tuning attributes have been modified to improve performance on larger systems. The default values set for `maxusers` and `vm-page-free-target` depend on the amount of memory in the system.

- Pipes

  A file-based pipes implementation replaces the socket-based pipes implementation for improved performance.

1.4 Packaging

Tru64 UNIX is available as a base system kit, containing the operating system, windowing environment, and documentation all integrated on CD-ROM, as well as the following extension, also included on the CD-ROM, that provide additional functionality and that require separate licenses and Product Authorization Keys (PAK) to access additional software packages, such as the optional Developer's Toolkit, Logical Storage Manager, Advanced Server for UNIX, and the Advanced File System Utilities.

1.4.1 Documentation

All Tru64 UNIX documentation produced by Compaq ships as PDF and HTML files on the Tru64 UNIX CD-ROM and as ASCII reference pages accessible from the `man` command. The following hardcopy documentation ships with the Tru64 UNIX CD-ROM:

- *Release Notes*
- *Installation Guide*
- *Installation Guide — Advanced Topics*
- *Update Installation Quick Reference Card*
- *Technical Overview*
- *Quick Reference Card*
- *Documentation Overview*

Complete hardcopy documentation and several third-party books are also available. For more information on the makeup of the documentation set, including optionally available documentation, see the *Documentation Overview*.

1.4.2 Licensing

Tru64 UNIX operating system software is furnished under the licensing of the Compaq Computer Corporation Standard Terms and Conditions.

In addition to the Operating System Base License, which is the prerequisite for all other licenses, there are five types of operating system licenses available on Alpha systems.

- Symmetric Multiprocessing (SMP) Extension to Base License
- Concurrent Use Licenses
- Unlimited Interactive User Licenses
- Hardware Partitioning License
- Single Servers License

For further information on these licenses, see the *Software Product Description*.

Tru64 UNIX provides the enabling technology to support static hardware partitions on certain servers only. These hardware partitions allow multiple instances of the operating system, which increases flexibility in testing new versions and running multiple versions for applications. Please consult the Systems and Options Catalog, located at http://www.digital.com/info/SOC for detailed configuration guidelines. The use of Tru64 UNIX in hardware partitions requires a Tru64 UNIX Hardware Partitioning License for each additional partition. For more information, please refer to the section titled "Software Licensing" in the *Software Product Description*.

1.4.3 Optional Software

There is a great deal of optional software available for Tru64 UNIX; this software includes additional programming and networking utilities as well as the TruCluster products, which maintain a cluster of processors.

- TruCluster Server Software

  This optional software product enables you to manage multiple systems in a cluster as a single system, significantly reducing the effort and complexity of cluster management. Applications are more highly available, more easily deployed, and scalable in a TruCluster Server environment.

- Logical Storage Manager

  The Logical Storage Manager (LSM) is an integrated, host-based solution to data storage management, providing concatenation, striping, mirroring, and a graphical user interface that allows data storage management to be done on line, without disrupting users or applications.

- Advanced File System Utilities

  The Advanced File System Utilities extend the high availability and flexibility of AdvFS. They provide a graphical user interface to ease management tasks and online utilities to resize file systems dynamically, balance the percentage of space used on volumes, undelete files, stripe files and clone files for hot backup.

- Developer's Toolkit

  The Developers' Toolkit is designed for programmers using languages other than C, like Fortran, C++, Ada, or Pascal, who require the complete software development environment but who do not require the C compiler, which is not available in this kit.

- Enterprise Toolkit for Visual Studio

  The Enterprise Toolkit is a set of extensions or add-ins to Microsoft Visual Studio that support developing C, C++, and Fortran applications for Tru64 UNIX servers. With Compaq Enterprise Toolkit for Visual Studio, developers can use the popular Microsoft Visual Studio tool to develop, edit, compile, build, and debug applications for Tru64 UNIX or Windows from a single desktop. You can create and manage basic UNIX applications, or create more powerful and complex client/server and distributed applications, harnessing the power of 64-bit Tru64 UNIX Alpha technology with a single set of PC tool. Additionally, the Compaq Enterprise Toolkit provides developers a rich set of performance and memory analysis tools. The Enterprise Toolkit uses Visual Studio's documentation browser, the HTML Help Viewer, to provide access to UNIX and product documentation from the same window that developers view Windows documentation.

- StorageWorks Software

  This cost savings package includes two key storage products: the Logical Storage Manager and Advanced File System Utilities. StorageWorks delivers high availability, configuration flexibility on line, optimal file system performance, and data protection.

- Advanced Server for UNIX

  Advanced Server for UNIX (ASU) provides seamless interoperability between Tru64 UNIX servers, Windows NT servers, and Microsoft Windows clients. ASU enables a Tru64 UNIX system to run the services that make it appear as a Microsoft Advanced Server.

- SCSI CAM Layered Components

  SCSI CAM Layered Components (CLC) provides device drivers and tools for two types of SCSI devices: robotic medium changers and magneto-optical disk drives, both read-write optical and write-once, read many (WORM).

- Prestoserve™

  Prestoserve is a disk write accelerator for disk block device write operations.

- Open3D

  This optional software provides a complete development and run-time environment for two- and three-dimensional applications. Open3D provides support for many graphical accelerators.

- Multimedia Services

  This optional software brings audio and video capabilities to Compaq Tru64 UNIX workstations, and provides a full multimedia programming library for developers.

- Source Materials Options

  A source kit is available for users who need to retrieve and modify selected source modules, primarily for making highly specialized modifications.

See the *Software Product Description* for additional information on any or all of these optional software packages.

2

System Management

Tru64 UNIX supports all the customary UNIX system administration utilities from OSF/1 Version 2.0, including the following:

- `tar`, `dump/restore`, and `dd` for performing backups and restores
- `df`, `du`, `scu`, `radisk`, and `disklabel` for managing disks and disk usage
- `tunefs`, `newfs`, and `fsck` for managing file systems
- `dbx` for debugging the kernel
- `adduser` for creating user accounts.

For more information on these utilities, see the *System Administration* guide; and the appropriate reference page for each utility.

In addition to providing system administration utilities from the OSF, Tru64 UNIX offers a number of other useful utilities, such as the following:

- Event Management (EVM)
- SysMan Tools
- Software Subset Management Utility `setld`
- Analysis Tools with Object Modification
- Enhanced Kernel Debugging
- Dynamically Loadable Subsystems
- Dynamic System Configuration
- Dynamic Device Recognition
- Dataless Management Services
- Monitoring Performance History
- Insight Manager
- Hardware Management
- Bootable Tape

This chapter describes these utilities.

2.1 Installation

Tru64 UNIX supports full and update installations either from a CD-ROM or across the network from a Remote Installation Services (RIS) server. For more information on RIS, see the guide *Sharing Software on a Local Area Network*.

2.1.1 Full Installation

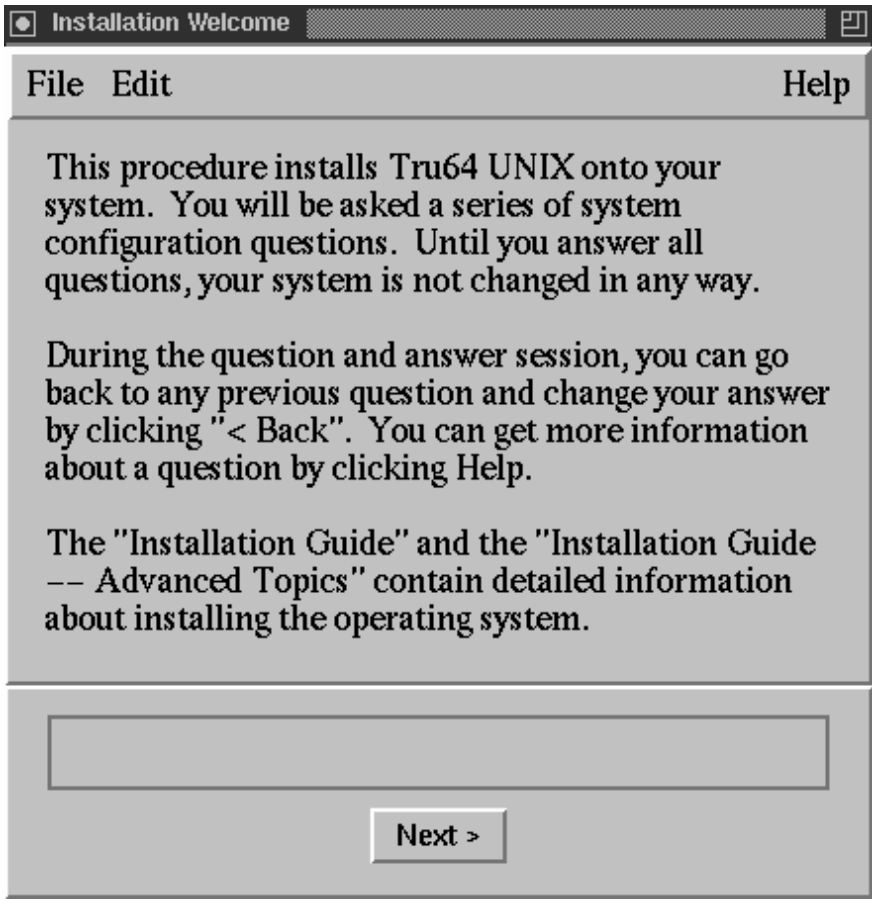
A Full Installation allows users to install Tru64 UNIX on new and existing systems. The Full Installation procedure offers a recommended file system layout, kernel option selection, and subset selection; however, you can also completely customize any or all operating system parameters.

You can use either a graphical interface or a text based interface to install the operating system quickly and easily. The graphical interface has a task like design that steps you through each phase of the setup process and lets you go backward and forward at any time. The text based interface also guides you through each setup phase and you can go back and change your answers, if necessary.

Figure 2–1 shows the initial window of the graphical Full Installation interface.

A Full Installation creates new file systems and swap space and overwrites existing system and user-created files on the disk partitions where the file systems and swap spaces are to be installed. You have the option to use preset default values for the disk layout and swap space allocation or to completely customize the locations of file systems and swap space.

Figure 2–1: Initial Window of Full Installation



The Full Installation provides the following features:

- Dynamic disk partitioning
- Simultaneous installation of the base operating system and Worldwide Language Support software
- Advanced File System (AdvFS)
- Ability to install and configure the Logical Storage Manager (LSM)
- Automatic resolution of software dependencies during optional software selection
- Ability to invoke user-supplied files at predefined invocation points to further customize the installation process
- Ability to clone the installation or configuration characteristics of one system to one or more similar systems

- Ability to identify physical disks to facilitate correct disk selection
- Access to a UNIX shell to perform file system and disk management tasks

You can customize and extend the Full Installation by creating custom scripts or programs to run at three process points during installation: before the installation begins, after software subsets load, and after the system reboots. The files you create can be loaded on a diskette, a CD-ROM, or a RIS server for use by the installation process.

A Full Installation creates a Configuration Description File (CDF) called `install.cdf` that can be used to replicate the Full Installation on other similar systems. You can also capture configuration data into a `config.cdf`file from a running system and replicate the following during a Full Installation:

- Network card and router configuration
- Domain Name Service (DNS)
- Network Information Service (NIS)
- Network Time Protocol (NTP)
- Printer services
- Mail services

For more information on Full Installations, see the *Installation Guide*. For more information on installation cloning, configuration cloning, and the use of user-supplied files, see the *Installation Guide — Advanced Topics*.

2.1.2 Update Installation

An Update Installation updates the operating system from Version 4.0D or 4.0F to Version 5.0. An Update Installation preserves the following:

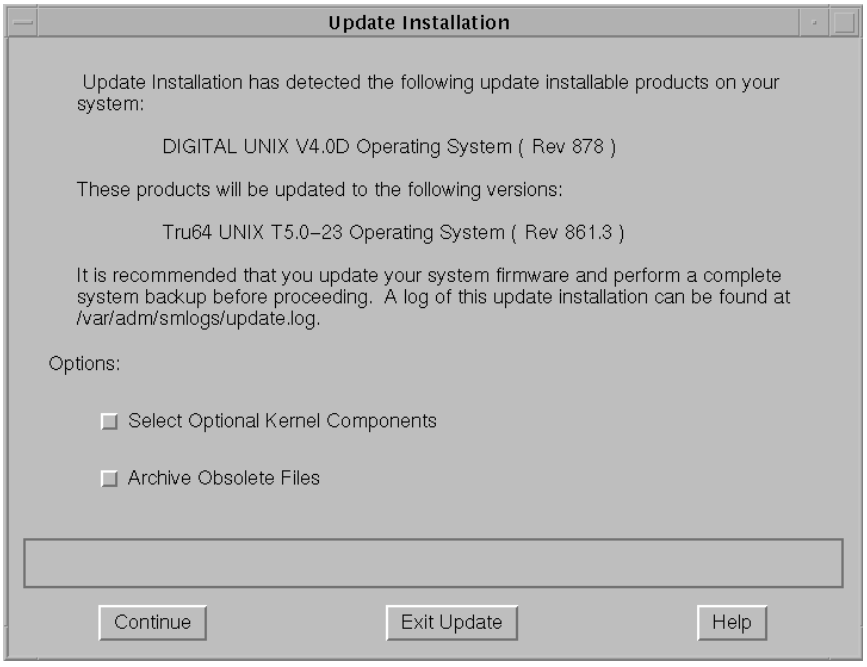
- Disk partitions
- File systems
- File customizations
- Network environment
- Print environment
- Mail environment
- User accounts
- User created files
- Other system setup you may have done

The software subsets that comprise the operating system are known as base software subsets. During an Update Installation, the base software subsets that are already installed on the current version of the operating system are updated to Version 5.0. In addition, any mandatory base software subsets that were introduced in Version 5.0 are installed automatically. Base software subset names start with the prefix `OSF`.

Worldwide Language Support (WLS) software subsets that are already installed on the current version of the operating system are updated to Version 5.0. In addition, any mandatory WLS software subsets that are introduced in Version 5.0 are installed automatically. WLS software subset names start with the prefix `IOS`.

Figure 2–2 shows the first window of the Update Installation procedure.

Figure 2–2: Initial Window of Update Installation



You cannot install additional optional software subsets during an Update Installation nor update layered products. You can, however, install additional optional software subsets by using the `setld` command when the Update Installation is complete. To update layered products, it may be necessary to delete the existing version and reinstall the new version that is designed to operate with Version 5.0. The Update Installation notifies you accordingly.

Do not perform an Update Installation if you want to change the type, location, or size of file systems or if you want to install optional software because those features are not offered during an update.

Table 2–1 summarizes the features of an Update Installation.

The Update Installation features are classified into two types: the features you control and the features that are built into the update process:

- The features that you control when you begin an Update Installation are shown in Table 2–1.
- The features that are built into the Update Installation process are shown in Table 2–2.

Table 2–1 shows the Update Installation features that you can turn on or off.

Table 2–1: User-Controlled Features of the Update Process

User Options	Description
Unattended Update Installation	If you do not need to select optional kernel components or archive obsolete files, you can invoke the Update Installation with the `-u` flag to run the update without any user intervention.
Kernel Component Options	You have the option to build either mandatory only or all kernel components into the kernel or you have the option to interactively select optional kernel components.
Archive Obsolete Files	You have the option to archive obsolete files before they automatically are removed by the Update Installation.

Table 2–2 describes the update features that are built into the update process automatically.

Table 2–2: Built-In Features of the Update Process

Built-In Feature	Description
Notification of conflicting layered products	Notifies you when an installed layered product may not be compatible with the new version of the operating system; this layered product may need to be reinstalled later.
Removes layered products that prevent the update from continuing	Upon your confirmation, removes layered products that prevent the update from continuing.
Updates base operating system and WLS software to new version	Updates existing installed subsets and installs new mandatory subsets introduced in the new version.

Table 2–2: Built-In Features of the Update Process (cont.)

Built-In Feature	Description
Checks for changed file types	Checks for file types that have been changed. The update might not be able to proceed if certain conflicts are found.
Disk space recovery	Provides the option to remove unnecessary software subsets and `.PreUPD`, `core` and extra kernel files to recover disk space if there is not enough file space to complete the update.
Executes instructions provided in user-supplied files	You can customize an Update Installation by creating and moving user-supplied scripts, programs, or executables to the appropriate location. If the update process finds files with the correct names in the appropriate locations, the files are executed.

For more information on the Update Installation, see the *Installation Guide* and the *Update Installation Quick Reference Card*.

2.1.3 The setld Utility

The `setld` utility allows system administrators to install software subsets, list installed subsets, and delete subsets that are formatted according to the guidelines set forth in *Guide to Preparing Product Kits*. For example, a system administrator might use the `setld` utility to install optional subsets that were not installed during a Full or Update Installation of the operating system.

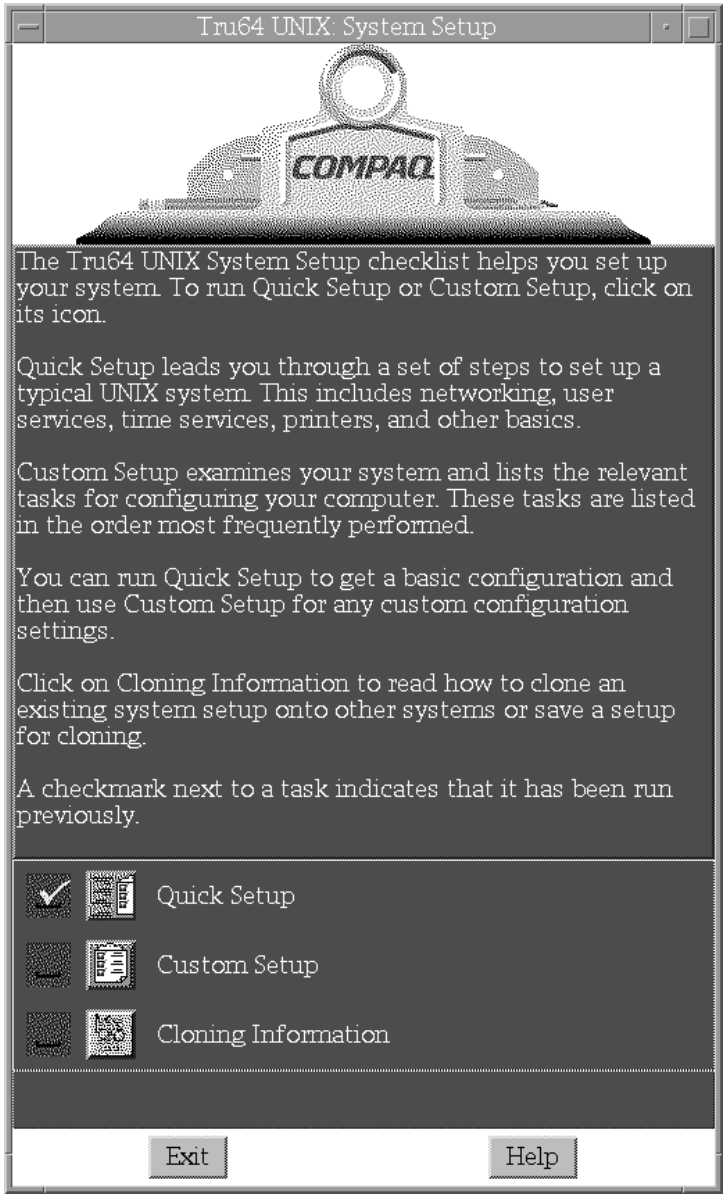
Application programmers should use the Compaq kitting process when packaging software subsets designed to be installed on Tru64 UNIX systems.

For more information on the `setld` utility, see the *Installation Guide* and the `setld`(8) reference page.

2.2 Configuration

Once the Tru64 UNIX software is installed, and if you have graphics capabilities, you can use SysMan System Setup to set up your system. System Setup enables you to invoke the Quick Setup and Custom Setup applications. Figure 2–3 illustrates the System Setup.

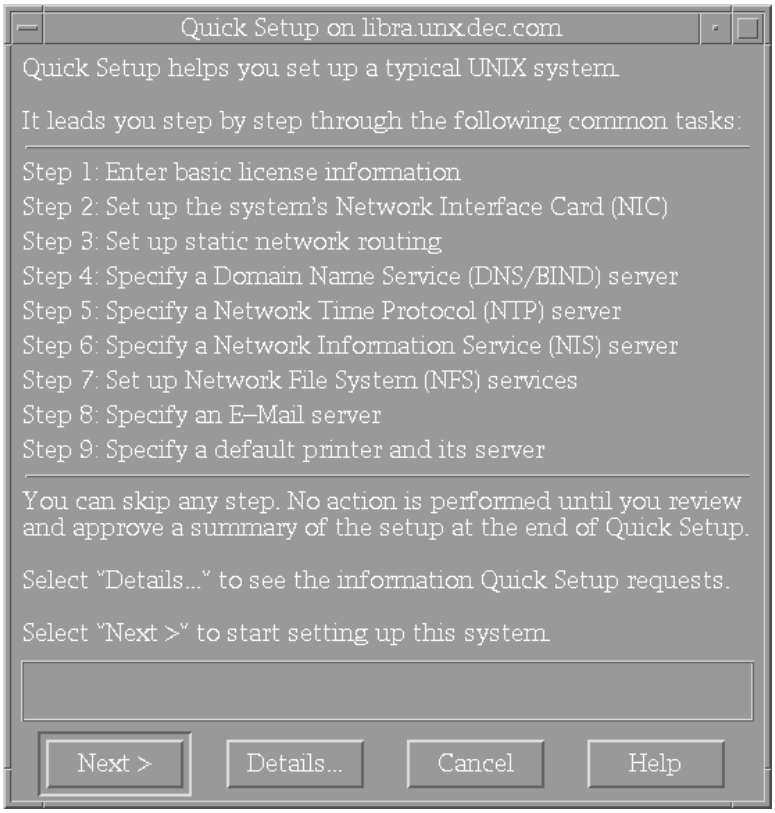
Figure 2–3: System Setup



Quick Setup has a wizard-like design that lets you enter a minimal amount of key information. Quick Setup updates your system with the basic configuration needed to get a client system up and running, including network connection, mail, and print capabilities. Quick Setup should satisfy the configuration needs for most systems. Even for systems that will

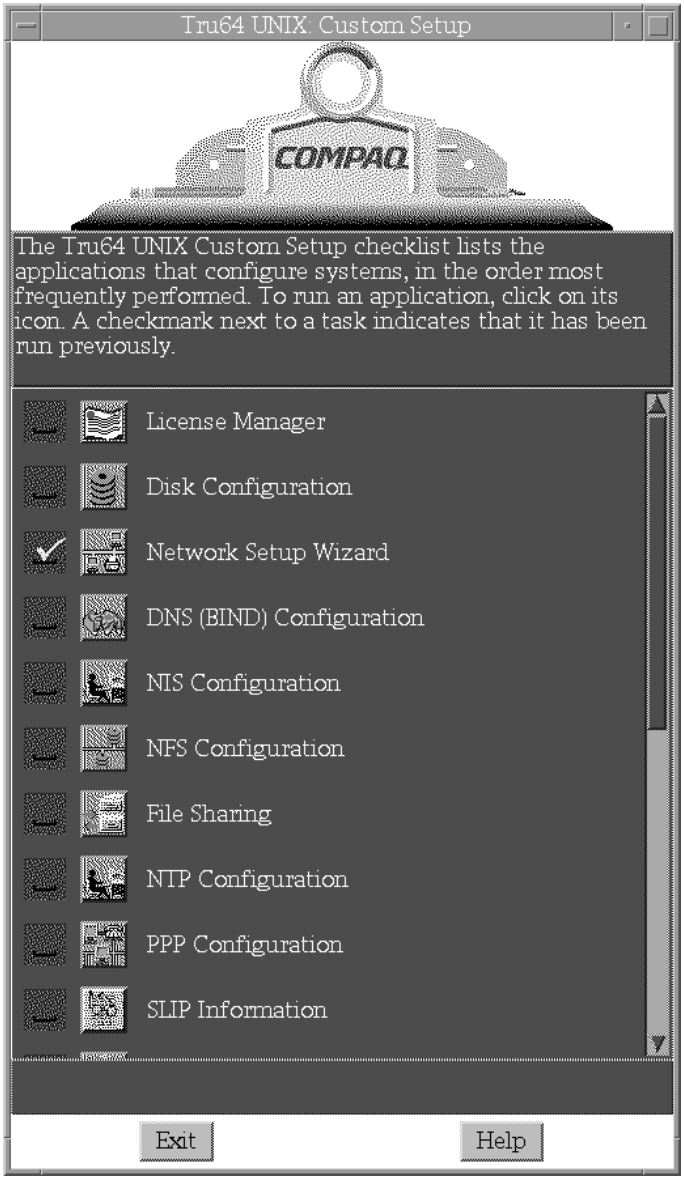
be configured as servers, it is recommended that you use Quick Setup first to configure your system then customize it with advanced applications in Custom Setup. Figure 2–4 illustrates Quick Setup.

Figure 2–4: Quick Setup



Custom Setup lets you perform sophisticated system configuration that is beyond Quick Setup's scope. When Custom Setup runs, it examines your system and presents a list of configuration applications that are relevant for your system; these applications are displayed as a checklist. Once you have accessed an application, a checkmark appears next to the application. For information about individual applications, see the Welcome to SysMan online help volume. Figure 2–5 illustrates Custom Setup.

Figure 2–5: Custom Setup



The first time you log in as `root` after a system installation or the first time you log in to a factory installed software (FIS) system, you will have the option to use either Quick Setup to configure a limited set of system parameters (including network and printer parameters) or use Custom Setup to set up your system for general use.

When you are logged in as superuser or `root`, you can invoke the SysMan System Setup at any time by clicking on the Configuration Checklist icon in the System_Administration folder, or entering the following command on the command line:

```
# /usr/sbin/checklist
```

The following are available from Custom Setup:

- Network Configuration Application
- BIND Configuration Application
- NIS - Network Information Service
- NFS Configuration Application
- License Manager
- Account Manager
- Mail Configuration Application
- Disk Configuration Application
- LAT - Local Area Transport
- UUCP - UNIX-to-UNIX Copy System
- NTP - Network Time Protocol
- Printer Configuration Application
- Security (BSD/2)
- Security Auditing
- Prestoserve I/O Acceleration
- Update Administration Application
- Graphical UI Selection Application

Many of the SysMan System Setup applicationare also available in textual interfaces that can be displayed on systems that only have character-cell displays.

For more information about system setup in general, see the *Installation Guide*, the *System Administration* guide, the *Network Administration* guide, the *Software License Management* guide, and the `setup`(8) reference page.

2.3 Storage Management

Storage Management consists of the Logical Storage Manager (LSM) and a graphical user interface to LSM, `lsmsa`. An optional license is required to access all the features described in this section.

2.3.1 Logical Storage Manager

Tru64 UNIX Logical Storage Manager provides host-based disk storage management using RAID technology. LSM is an enhanced port of the Volume Manager product developed by VERITAS.

The operating system includes the following basic features of LSM:

- Disk spanning (concatenation)

  Enables the combination of multiple physical disks into a single, larger virtual disk (volume) for use by large file systems or databases. For example, two disks could be concatenated to store the `/usr` file system.

These features and the graphical user interface management tools require an optional license that is included with the StorageWorks Software package.

- Data Striping (RAID 0)

  Improves disk I/O performance by interleaving data within a volume across several physical disks.

- Data Mirroring (RAID 1)

  Protects against data loss due to hardware malfunction by creating a mirror (duplicate) image of important file systems and databases. For maximum availability, data can be mirrored across different I/O controllers as well as different disks.

- Data Striping and Mirroring (RAID 0+1)

  Provides both improved system performance and high data availability.

- Comprehensive on-line storage management capabilities

  Includes convenient interfaces and utilities to add, move, modify, remove, and manage storage space and volumes without rebooting or interrupting users. For example, an LSM volume containing an AdvFS file system or swap space can be mirrored or moved to a different physical disk without stopping I/O to the volume. The `mkfdmn` and `mkfset` utilities are used to create AdvFS file systems; the `vdump` and `vrestore` utilities provide backup support.

In addition, Tru64 UNIX LSM supports the following new features:

- RAID-5

  Provides high data availability and improved I/O performance by storing parity information along with striped data.

- Hot-sparing

  Provides automatic reaction to I/O failures on redundant (mirrored or RAID-5) objects by relocating affected objects to spare disks or other free space.

- Dirty-region logging (replacement for block-change-logging)

  Tracks regions on a mirrored volume that have changed due to I/O writes. Provides speedy recovery of a mirrored volume after a system failure, by restoring only the changed regions.

- Automatic configuration database load-balancing

  Maintains an optimal number of LSM configuration databases in appropriate locations without manual intervention

The LSM disk concatenation capabilities are included as part of the Tru64 UNIX Operating System.

For each logical volume defined in the system, the LSM volume device driver maps logical volume I/O to physical disk I/O. Users can administer LSM by using either a set of command-line utilities or one of two Motif-based graphical interfaces, `dxlsm` and `lsmsa`.

To help users migrate their existing UFS or AdvFS partitions into LSM logical volumes, LSM includes a utility that will transform each partition in use by UFS or AdvFS into a nonstriped, nonmirrored LSM volume. After the transformation is complete, administrators can mirror the volumes if they want to do so.

LSM volumes can be used in conjunction with AdvFS (as part of an AdvFS domain) and with RAID disks. For more information on LSM, see the *Logical Storage Manager* guide.

2.3.2 Storage Administrator

The Storage Administrator, `lsmsa`, is a graphical user interface that uses the Java run-time environment. Its purpose is two-fold: it provides a method of invoking LSM commands and it monitors LSM file status. When the main window is displayed, a hierarchical view of LSM objects is presented. Clicking on an object displays the objects of that type and a table of information about them.

To use this utility, you must be the root user or have the appropriate permissions. In addition, you must register the Advanced File System Utilities license.

2.3.3 Prestoserve File System Accelerator

The Prestoserve file system accelerator is a hardware option for machines running Tru64 UNIX. Prestoserve speeds up synchronous disk writes on a local client system and on an NFS server by reducing the amount of disk I/O on these machines. Prestoserve does this by caching frequently written data blocks in its nonvolatile memory and then writes them asynchronously to disk. Prestoserve also caches UFS and AdvFS metadata the same way.

The software required to drive the Prestoserve hardware is an optional subset in Tru64 UNIX. Once the subset is installed, it can be enabled with a Product Authorization Key (PAK) that is included with the hardware.

2.4 System Management Tools

SysMan is the suite of graphical applications developed for Tru64 UNIX system and network administrators. The SysMan applications are launched from the CDE Application Manager and have supported command line counterparts.

These are the SysMan tools that have been developed to complement the current SysMan suite of applications:

- The SysMan Station
- The SysMan Menu

SysMan makes your job as a system administrator easier by providing you with a graphical user interface for each of your administration tasks, such as installation, configuration, daily administration, monitoring, kernel and process tuning, and storage management. These applications can be accessed through the System_Admin folder in the CDE Application Manager.

Although the SysMan tools were designed to take advantage of the Common Desktop Environment (CDE), most applications will work outside of CDE with other window or display managers. Users who are not running CDE can access the utilities individually by invoking them from the command line, provided the `DISPLAY` environment variable is properly set on their systems. For instance, the following command invokes the Network Setup Wizard, which configures a system to operate in a network:

```
# netconfig
```

Many of the SysMan tools are also available in text-based interfaces that can be displayed on systems that only have character-cell displays.

2.4.1 The SysMan Station

The SysMan Station provides a high profile view and status of a system's physical and logical objects. It is intended to be the central point from which to manage a Tru64 UNIX system. SysMan Station launches other SysMan tools to perform the tasks. The SysMan Station lets you monitor a system, group of systems, or an entire cluster and administer system resources. You can launch the SysMan or invoke applications directly from the Tools menu in the SysMan Station. It can run on a standard Java capable display (such as a UNIX workstation) or within a PC's browser, or it can be downloaded and run directly on a PC. Use SysMan Station to:

- Monitor the status of a system or cluster
- Display detailed information about a system or cluster
- Provide a single location for management activity
- Display events and track events that lead to a problem

The SysMan Station is launched directly from an icon on the CDE front panel or from the command line by entering `/usr/sbin/sms`. The front panel control is shown in Figure 2–6.

Figure 2–6: SysMan Station Control on CDE Front Panel

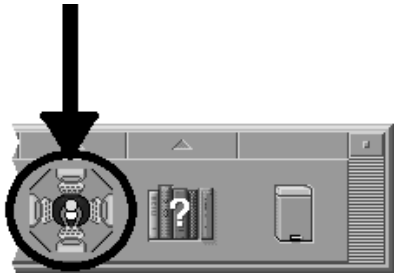
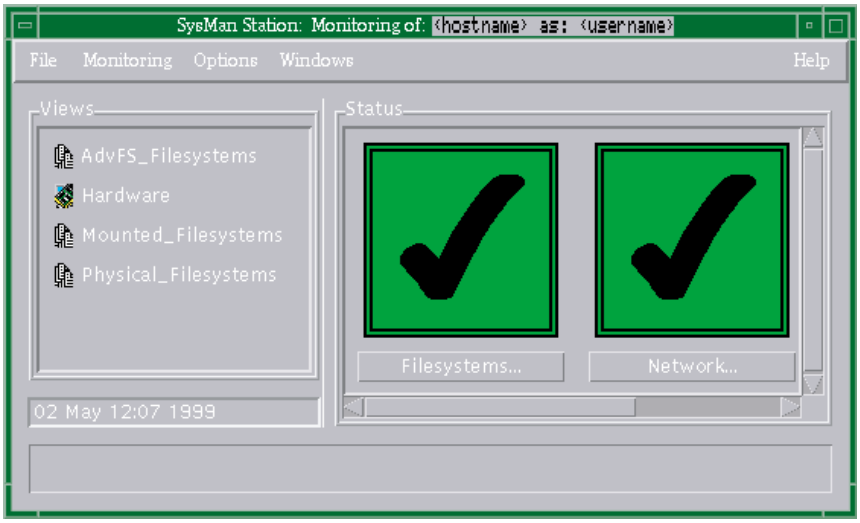


Figure 2–7 illustrates the SysMan Station.

Figure 2–7: The SysMan Station



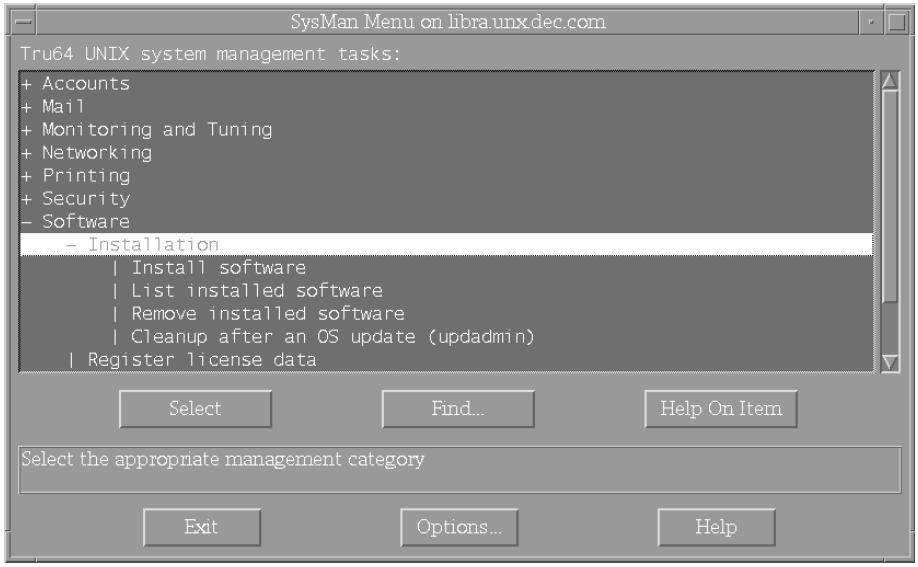
2.4.2 The SysMan Menu

The SysMan Menu provides a menu of system management tasks in a tree-like hierarchy with branches of general categories, and leaves for actual tasks. Users can expand or contract a branch to show the subbranches and leaves within a main branch. Selecting a leaf invokes a task, which opens a dialog box for performing the task. The categories comprise Accounts, Mail, Monitoring and Tuning, Networking, Printing, Security, Hardware, Software, Storage, Support and Services, and General Tasks.

The SysMan Menu is invoked from an icon in the CDE Application Manager or directly from the command line by entering `/usr/sbin/sysman`. The SysMan Menu can be used in several different user environments: X Windows, Web browers, or simple character terminal displays.

Figure 2–8 illustrates the SysMan Menu.

Figure 2–8: The SysMan Menu



2.4.3 SysMan Applications

SysMan offers the following system administration applications, but the contents of each folder in SysMan can vary depending on which additional software subsets you have installed:

- Configuration Checklist

  You can use these applications to set up your system for general use after Tru64 UNIX has been installed. See Section 2.2 for more information.

- Configuration Applications

  Use these applications to configure network and system settings:

  – Network configuration (`netconfig`)

  – BIND configuration (`bindconfig`)

  – NFS configuration (`nfsconfig`)

  – Mail configuration (`mailconfig`)

  – Print configuration (`printconfig`)

  – Disk configuration (`diskconfig`)

  – LAT configuration (`latsetup`)

  – NIS configuration (`nissetup`)

- Daily Administration Applications

  After a system has been configured, use these applications to perform routine administrative tasks:

  – Account Manager (`dxaccounts`)
  – Archiver (`dxarchiver`)
  – File Sharing (`dxfileshare`)
  – Host Manager (`dxhosts`)
  – License Manager (`dxlicenses`)
  – Shutdown Manager (`dxshutdown`)
  – Audit Manager (`dxaudit`)
  – Power Management (`dxpower`)
  – DHCP Configuration (`dhcpconf`)
  – Display Window (`dxdw`)

- Monitoring and Tuning Applications

  While a system is running, use these applications to monitor and tune its resources:

  – Insight Manager Agents (IMXE)
  – System Information (`dxsysinfo`)
  – Kernel attributes (`dxkerneltuner`)
  – Monitor and manage processes (`dxproctuner`)

- Storage Management Applications

  Use these applications to configure a running system's file systems:

  – Bootable Tape Creation (`btcreate`)
  – Prestoserve (`dxpresto`)
  – Logical Storage Manager (`dxlsm`)

- Tools

  Use these applications to check the status of the system:

  – Network Statistics (`netstat`)
  – Virtual Memory Statistics (`vmstat`)
  – I/O Statistics (`iostat`)
  – Current users (`who`)

For more information on the SysMan tools, please click on the Welcome to SysMan icon in the System_Admin folder.

2.4.4 The CDE Application Manager

If you are running Tru64 UNIX with the CDE Desktop, you have access to the CDE Application Manager. The `System_Admin` group on the CDE Application Manager launches the following SysMan tools:

SysMan Configuration Checklist	Opens the System Setup checklist (which contains Quick Setup and Custom Setup) to enable complete access to system configuration applications
SysMan Menu	Opens the SysMan Menu
SysMan Station	Opens the SysMan Station
Welcome to SysMan	Opens the SysMan Overview help volume, which describes the Tru64 UNIX system management tools and applications

The System_Admin group also contains the following categories of system administration groups:

- The Configuration group contains applications for configuring and setting up a Tru64 UNIX system after it has been installed. Once a system has been set up, it is unlikely that an administrator would have to use these applications on a regular basis. From this group you can launch these applications:
  - BIND/DNS
  - CDE Setup
  - DHCP Server
  - Disk Config
  - Mail Config
  - NFS Config
  - NIS Setup
  - PPP Config
  - Print Config
  - SLIP
  - latsetup
- The Daily Admin group contains applications that system administrators use to perform typical administration tasks on a daily basis. From this group you can launch these applications:

  - Account Manager
  - Archiver
  - Audit Manager
  - Display Window
  - Event Viewer
  - File Sharing
  - Host Manager
  - License Manager
  - Mail User Administration
  - Power Management
  - Shutdown
  - System Information
- The Monitoring Tuning group offers applications to tune and monitor the system once it is up and running. From this group you can launch these applications:
  - Kernel Tuner
  - Process Tuner
  - System Information
  - Insight Manager Agents
- Software Management contains applications that manage and install additional software on the system. From this group you can launch these applications:
  - Update Installation Cleanup
  - Update Installation
  - Install Software
  - Delete Software
  - List Software
- Storage Management contains applications that help administrators manage and monitor file systems. From this group you can launch these applications:

- Advanced File System
- Bootable Tape
- Logical Storage Manager
- Prestoserve I/O Accelerator

2.5 Event Management

The Event Manager (EVM) provides a single point of focus for the multiple channels through which system components report event and status information, by combining events from all sources into a single event stream. The system administrator can either monitor the combined stream in real time or view historical events retrieved from storage. EVM's viewing facilities include a graphical event viewer, which is integrated with the SysMan application suite, and a full set of command line utilities, which allow events to be filtered, sorted, and formatted in a variety of ways.

The system administrator can also configure EVM to perform automatic notification of selected conditions. Rather than replacing the familiar event channels, such as syslog and binlog, EVM encapsulates them, so these channels remain in place and continue to handle the same set of events. At the same time, however, the introduction of EVM makes them much more accessible.

Key features of EVM are:

- Centralized event information
- Facilities for users and applications to post and monitor events
- Continued support for all existing event channels, including `syslog` and the binary error logger
- Support for encapsulation of custom event channels
- Integration with DECevent for translation of binary error log events
- Graphical event viewer, fully integrated with the SysMan application suite
- Choice of summary-line or detailed view of events, including on-line explanations
- Full set of command line utilities for posting and handling events from shell scripts and from the command line
- Configurable event logger allows full control over which events are logged and optimizes storage space used by identical events
- Configurable event forwarding allows automatic notification of selected events

- Automatic logfile management performs daily archiving and purging tasks
- Supporting application programming interface (API) library
- Configurable authorization for posting or accessing events

For more information on the Event Manager, see *System Administration*.

2.6 DECevent Translation and Reporting Utility

DECevent is an error log formatting utility for Tru64 UNIX that translates system event log files into formatted ASCII reports. DECevent supports both a command-line and a graphical user interface (GUI).

DECevent is able to format newer record types that are unknown to `uerf`.

Event report information can be filtered by event types, date, time, and event entry numbers. Event report formats can be selected from full disclosure to very brief information messages. The `-i` (include) and the `-x` (exclude) options provide a wide range of selection criteria to narrow down the focus of event searches.

The DECevent utility also offers an interactive command shell interface, accessible with the `-int` option, that recognizes the same commands used at the command line. From the interactive command shell users can customize, change, or save system settings.

DECevent uses the system event log file `/usr/adm/binary.errlog` as the default input file for event reporting.

Tru64 UNIX users need superuser privileges to use DECevent.

For more information on DECevent, see the *DECevent Translation and Reporting Utility* manual.

2.7 Dynamically Loadable Subsystems

Tru64 UNIX provides the ability to package, load, and manage kernel subsystems on Tru64 UNIX systems.

Instructions on how to write and package loadable device drivers so that they will install and execute on Tru64 UNIX systems are discussed in the Device Driver documentation. *Writing Kernel Modules* explains how to write and package loadable kernel subsystems so that they too will install and execute on Tru64 UNIX systems. The *Programmer's Guide* also discusses in some detail the framework that supports the dynamic configuration and tuning of kernel attributes.

2.8 Dynamic System Configuration

To simplify system tuning, Tru64 UNIX allows you to change certain kernel attributes without having to edit the the system configuration file or the file `param.c`, and without having to rebuild and reboot a target kernel for the changes to take effect. Through the use of **attribute tables**, each kernel subsystem —– whether a Tru64 UNIX kernel subsystem or one developed by a third-party vendor —– can define kernel attributes that can be changed at run time by using the `/sbin/sysconfig` command with the `-r` option (if the kernel attribute supports run-time reconfiguration), or at boot-time by adding or modifying entries in the kernel attribute database, `/etc/sysconfigtab`, and rebooting.

For more information, see the *System Administration* guide; and the *System Configuration and Tuning* guide.

Kernel attributes can also be modified with the `dxkerneltuner`; for further information, see the `dxkerneltuner`(8) reference page.

2.9 Dynamic Device Recognition

Dynamic Device Recognition is a framework for describing the operating parameters and characteristics of SCSI devices to the SCSI CAM I/O subsystem. You can use DDR to include new and changed SCSI devices into your environment without having to reboot the operating system.

DDR is preferred over the static methods for recognizing SCSI devices, because DDR will not disrupt user services and processes as happens with static methods of device recognition.

2.10 Dataless Management Services

Tru64 UNIX supports dataless management services (DMS), which allows the / (root), `/usr`, and `/var` partitions of a system to live on a DMS server and be NFS mounted over the network by a DMS client. The / and `/var` partitions are unique to each DMS client, while `/usr` is shared. The DMS client swaps and dumps locally. Additional file systems can be mounted using NFS.

DMS reduces disk needs and simplifies system administration, because administrators can administer and backup their DMS clients on the DMS server.

2.11 Monitoring Performance History

The Monitoring Performance History (MPH) utility gathers timely and accurate information on the reliability and availability of the Tru64 UNIX operating system and associated platforms. For more information, see the *Installation Guide*.

2.12 Insight Manager

This release provides Web Based Enterprise Management (WBEM) capabilities by integrating the Compaq Insight Manager Agents into Tru64 UNIX. Insight Manager enables Web-based browsing and monitoring of local and remote system hardware and software resources. You can activate these Web browsing features from a dedicated HTTP port, or from the Compaq Insight Manager CIM32 or CIMXE Management Consoles running on Compaq NT servers. Insight Manager Agents for Tru64 UNIX includes SNMP based subagents and WBEM capabilities to present SNMP data in a format viewable by a Web browser. It provides a rich view of the data using HTML 2.0 and JavaScript in the form of Web pages.

The Insight Manager Device home page provides three service icons called "Compaq Insight Management Agents", "Compaq Tru64 UNIX Configuration Report", and "Compaq Tru64 UNIX SysMan". Selecting the Compaq Insight Management Agents icon displays the device pages of the target system.

The home page also presents three hyperlinks: 1) Refresh, 2) Options, and 3) Devices. The Options link provides various Agent configuration options. The Devices link provides auto-discovery of other Insight Manager Agents and Insight manager XE Servers running on the local network, allowing direct access to their 2301 ports.

The Compaq Insight Management Agents page consists of three frames. The upper left frame displays a conditions legend and the following links:

- Agent Help (Use this link for online help)
- Summary (Use this link to navigate back to the Summary Page)
- Device Home (Use this link to return to the Device home page)
- Options (Use this link to navigate to the Options Page and set options for Display Mode, Help Icons, and AutoRefresh intervals)

The lower left frame is a navigational frame that provides links to various system components grouped under Configuration, Mass Storage, NIC

(Networking), Utilization, and Recovery. Clicking on the system component links displays component specific data in the main pane window.

2.13 Hardware Management

This release introduces the `hwmgr` utility for hardware management. This utility lets you manage hardware components including disk and tape drives, processors, and buses.

The `hwmgr` utility offers a wide variety of options, including the following:

`-view`	Displays information. The variations on this option include `-view devices`, which provides the hardware identifier, device special file name, model, and location of all the devices on the system. Another variation, the `-view hierarchy` option, displays the current hardware component hierarchy.
`-flash`	Identifies a disk by flashing its LED. The disk can be identified by its SCSI bus number, SCSI target number, logical unit number, or its device special file name.
`-show component`	Displays hardware components, including those that were previously unregistered but may not be currently registered with hardware management. This command returns a series of one-character flags indicating the component is currently registered, has device nodes associated with it, has a cluster wide unique name, has saved attributes associated with it, is inconsistent with the hardware component database.

For a full description of this utility, see the `hwmgr`(8) reference page.

2.14 Bootable Tape

Tru64 UNIX introduces the ability to create a standalone bootable tape of the operating system using a graphical user interface or command line options.

You can boot from the bootable tape as easily as you can boot from a CD-ROM or a RIS area, but without the overhead of selecting or installing subsets.

When you restore your system from the bootable tape, you must reconfigure your system using the System Management applications.

See the `bttape`(8), `btcreate`(8), and `btextract`(8) reference pages for more information.

3

Networking

This chapter describes the components of the operating system that enable a wide variety of networking capabilities.

3.1 Overview

The networking components of Tru64 UNIX come primarily from OSF/1 Version 1.0, although certain modules, like System V Release 4.0 STREAMS, which was unavailable in OSF/1 Version 1.0, were taken from the OSF/1 Version 1.2 code base. Some functions, like IP Multicasting and the packet filter applications, were taken from the public domain, enhanced, and integrated into the operating system as a service to our customers.

The Network File System (NFS) code, as well as the Remote Procedure Calling (RPC) code, Network Information Service (NIS), and remote daemons and their corresponding commands came from Sun Microsystem's Open Network Computing (ONC) Version 4.2.

Functions that Compaq has licensed and enhanced, such as Yellow Pages and the Network Information Service, which was licensed from Sun Microsystems, were ported to Tru64 UNIX from ULTRIX, Compaq's earlier version of the UNIX operating system. Although conforming to the OSF/1 Version 1.2 standards, these components were determined to be more robust than those in the corresponding code from the OSF.

Like all subsystems in the operating system, the networking subsystem is designed to provide a standardized programming interface to enable third-party vendors to develop and port their networking applications with a minimum of difficulty. To this end, this version of the operating system supports the following:

- Internet Protocol Suite
  - Application Protocols
    - Domain Name Service (DNS)
    - Routing protocols (RIP, OSPF, EGP, BGP, and DVMRP)
    - File Transfer Protocol (FTP)
    - Network File System protocol (NFS)

    - Telnet protocol (TELNET)
    - Trivial File Transfer Protocol (TFTP)
    - Finger protocol (FINGER)
    - Simple Mail Transfer Protocol (SMTP)
    - Simple Network Manager Protocol (SNMP)
    - Post Office Protocol (POP3)
    - Internet Message Access Protocol (IMAP4)
  - Transport protocols
    - User Datagram Protocol (UDP)
    - Transmission Control Protocol (TCP)
  - Network level protocols
    - Internet Protocol (IP)
    - Address Resolution Protocol (ARP)
    - Internet Control Message Protocol (ICMP)
    - Internet Group Management Protocol (IGMP)
    - Point-to-Point Protocol (PPP)
    - Serial Line IP (SLIP) and Compressed Serial Line IP (CSLIP)
    - Resource Reservation Protocol (RSVP) RFC 2205
- Supported transports
  - Asynchronous Transport Mode (ATM)
  - Ethernet
  - Fast Ethernet
  - Gigabit Ethernet
  - Fiber Distributed Data Interface (FDDI)
  - Token Ring
  - Multiple Adapter Support
- Application programming interfaces
  - X/Open Transport Interface (XTI/TLI)
  - BSD 4.3 Sockets, BSD 4.4 Sockets, XNS 4.0 and XNS 5.0 Sockets
  - System V Release 4.0 STREAMS
  - Data Link Interface (DLI)
  - Data Link Provider Interface (DLPI)

  - Extensible SNMP (eSNMP)
  - UNIX 98Sockets
  - IPv6 Basic API (RFC 2553)
  - RSVP API (RAPI)
- Network administration software
  - The entire suite of network commands and utilities from OSF/1 Version 1.2
  - Ethernet, FDDI and loopback, packet filtering
  - Several popular packet filter applications from the public domain (`rarpd`, `tcpdump`, `tcpslice`, `nfswatch`, `nfslogsum`)
  - The `screend` security policy daemon
  - UUCP from HoneyDanBer
  - Local Area Transport (LAT)
  - Dynamic Host Configuration Protocol (DHCP)
  - NetRAIN
  - Network Interface Failure Finder facility
- Naming services
  - Berkeley Internet Name Domain (BIND)
  - Yellow Pages/Network Information Services (YP/NIS)
- Time services
  - Network Time Protocol (NTP)
  - Time Synchronization Protocol (TSP)

The following sections briefly discuss the networking functionality in this release of the operating system.

3.2 The Internet Protocol Suite

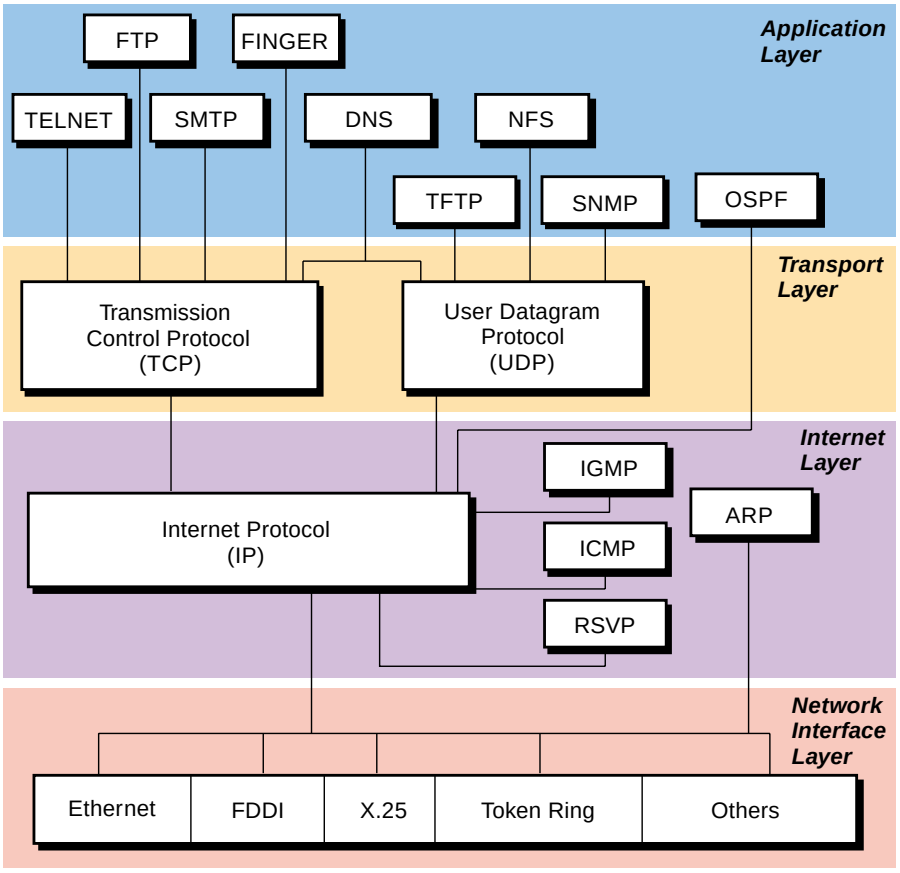
TCP/IP supports a suite of protocols, each of which provides a different service. These protocols allow networking communications to be independent of network hardware. The TCP/IP protocol suite is organized into the following groups:

- Application-level protocols, such as DNS, routing protocols (DVMRP, EGP, BGP, RIP, and OSPF), File Transfer Protocol (FTP), FINGER, TELNET, Trivial File Transfer Protocol (TFTP), Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP)

- Transport-level protocols, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP)
- Network-level protocols, such as Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and Internet Protocol (IP)

Figure 3–1 illustrates the relationship of the major protocols in the TCP/IP suite.

Figure 3–1: TCP/IP Protocols



ZK-0819U-AI

Applications programs send messages (streams or blocks of data) to the internet transport-level protocols, which are the UDP and the TCP. These protocols receive the data from the application, divide it into packets, add a transport header, and then pass the packets along to the next protocol layer, the Internet layer.

The Internet layer encloses the packet in an IP datagram, adds the datagram header, decides where to send the datagram (either directly to a destination or else to a gateway), and passes the datagram on to the network interface layer. The network interface layer accepts IP datagrams and transmits them as frames over a specific network hardware.

Frames received by a network go through the protocol layers in reverse. Each layer strips off the corresponding header information until the data is back at the application level. Frames are received by the network interface layer (for example, an Ethernet adapter), which strips off the physical layer header and sends the datagram to the internet layer. In the internet layer, the internet protocol strips off the IP header and sends the packet to the transport layer. The transport layer strips off the TCP or UDP header and sends the data up to the application layer.

3.2.1 Application-Level Protocols

When an application needs to send data to an application on another host, the application sends the information through transport level protocols to prepare the information for transmission. These protocols include DNS, EGP, BGP RIP, OSPF, FTP, NFS, TELNET, TFTP, FINGER, SMTP, and SNMP.

3.2.1.1 Domain Name Service Protocol

The Domain Name Service (DNS) allows one or more hosts in a domain to act as a name server for other hosts within the domain. DNS uses UDP or TCP as its underlying protocol and allows a local network to assign host names within its domain independently from other domains. UDP is the preferred protocol for use with DNS; however, if the UDP response is truncated, TCP can be used.

In the Tru64 UNIX environment, the Berkeley Internet Name Domain (BIND) naming service uses the Domain Name Service protocol. In this hierarchical naming system, local resolver routines may resolve Internet names and addresses using a local name resolution database maintained by the `named` daemon. If the name requested by the host is not in the local database, the resolver routine or the local `named` daemon queries the remote DNS name server.

3.2.1.2 Routing Protocols

Routing Protocols allow systems on either internal or external LANs to share routing information. In addition to the External Gateway Protocol (EGP), this version of the operating system supports the Border Gateway Protocol (BGP) and both the Routing Information Protocol (RIP) and Open

Shortest Path First Protocol (OSPF) as part of the `gated` v3.5A11 routing daemon from the Merit GateD Consortium (for more information on the `gated`, see Section 3.5.6). Refer to the Tru64 UNIX Software Product Description for a list of officially supported routing protocols.

Exterior Gateway Protocol (EGP)

The Exterior Gateway Protocol (EGP) allows the exterior gateway of an autonomous system to share routing information with exterior gateways on other autonomous systems.

An autonomous system is a group of networks and gateways for which one administrative authority has responsibility. Gateways are interior neighbors if they reside on the same autonomous system and exterior neighbors if they reside on different autonomous systems. Gateways that exchange routing information using EGP are said to be EGP peers (neighbors). Autonomous system gateways use EGP to provide reachability information to their EGP neighbors.

EGP allows an exterior gateway to provide remote communications among systems as follows:

- Ask another exterior gateway to agree to exchange reachability information
- Continually check to ensure that its EGP neighbors are responding
- Allow EGP neighbors to exchange reachability information by passing routing update messages

EGP restricts exterior gateways by allowing them to advertise only those destination networks reachable entirely within that gateway's autonomous system. Thus, an exterior gateway using EGP passes on information to its EGP neighbors, but does not advertise reachability information about its EGP neighbors.

EGP does not interpret the distance metrics that appear in routing update messages from other protocols. EGP uses the distance field to specify whether a path exists (a value of 255 means that the network is unreachable). The value cannot be used to compute the shorter of two routes, unless those routes are both contained within a single autonomous system. For this reason, EGP cannot be used as a routing algorithm. As a result, there is only one path from an exterior gateway to any network.

EGP routes are predetermined in the `/etc/gated.conf` file. This contrasts with the Routing Information Protocol (RIP), which can be used within (that is, interior to) an autonomous system of Internet networks that dynamically reconfigure routes. EGP assumes that IP is the underlying protocol. See the `gated`(8) reference page for further information.

Border Gateway Protocol

The Border Gateway Protocol (BGP) is an exterior routing protocol used for exchanging routing information between autonomous systems that are either multiple transit autonomous systems or transit and stub autonomous systems. BGP is related to EGP but operates with more capability, greater flexibility, and less required bandwidth. For example, BGP uses path attributes to provide more information about each route and, unlike EGP, maintains an Autonomous System (AS) path, which provides enough information (such as the AS number of each autonomous system the route has traversed) to prevent routing loops in an arbitrary topology.

Like EGP, BGP supports both internal and external sessions. When sending routes to an external peer, BGP prepends the local AS number to the AS path so that routes received from an external peer are guaranteed to have the AS number of that peer at the start of the path.

Routes received from an internal neighbor will not in general have the local AS number prepended to the AS path, and in general have the same AS path that the route had when the originating internal neighbor received the route from an external peer. Routes with no AS numbers in the path may be legitimately received from internal neighbors; these indicate that the received route should be considered internal to your own AS.

The implementation of BGP in this release of the operating system supports three versions of the BGP protocol (versions 2, 3, and 4). BGP versions 2 and 3 are quite similar in capability and function. They will only propagate classed network routes, and the AS path is a simple array of AS numbers. BGP 4 will propagate fully general address-and-mask routes, and the AS path has some structure to represent the results of aggregating dissimilar routes.

Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is an implementation of a distance-vector, or Bellman-Ford routing protocol for local networks and in Tru64 UNIX is contained in the `gated` daemon from the Merit GateD Consortium. RIP classifies routers as active and passive: active routers advertise their routes to other routers; passive routers listen and update their routes based on the advertisements they receive, but do not advertise themselves. Typically, routers run RIP in active mode, while hosts use passive mode.

A router running RIP in active mode broadcasts updates at set intervals. Each update contains paired values, where each pair consists of an IP network address and an integer distance to that network. RIP uses a hop count metric to measure the distance to a destination. The number of hops

along a path from a given source to a given destination refers to the number of gateways that a datagram would encounter along that path.

For example, a router advertises directly connected networks as having a hop count of one. Networks that are reachable through another gateway are two hops away, networks that are reachable through two gateways are three hops away, and so forth. Then RIP chooses the path with the shortest hop count.

Of course, using hop counts to calculate shortest paths between networks may not always produce optimal results. For example, a path with a hop count of three that crosses three Ethernet connections may be substantially faster than a path with a hop count of 2 that crosses two slow-speed serial lines. To compensate for differences in network and serial line rates of transfer, administrators can configure RIP routers to advertise artificially high hop counts for slow links.

Open Shortest Path First

Open Shortest Path Routing (OSPF) is a shortest path first (SPF) or link-state interior gateway protocol that distributes routing information between routers in a single autonomous system. Suitable for complex networks with a large number of routers, OSPF provides equal cost multipath routing whereby packets to a single destination can be sent by more than one network interface simultaneously.

A link-state protocol dictates that each router maintains a database describing the entire AS topology, which it builds out of the collected link state advertisements of all routers. Each participating router distributes its local state (that is the router's usable interfaces and reachable neighbors) throughout the AS by flooding. Each multiaccess network that has at least two attached routers has a designated router and a backup designated router. The designated router floods a link state advertisement for the multiaccess network and has other special responsibilities. The designated router concept reduces the number of adjacencies required on a multiaccess network.

OSPF allows networks to be grouped into areas. Routing information passed between areas is abstracted, potentially allowing a significant reduction in routing traffic. OSPF uses four different types of routes, listed in order of preference:

- intra-area
- inter-area
- type 1 external
- type 2 external.

Intra-area paths have destinations within the same area.

Inter-area paths have destinations in other OSPF areas and Autonomous System External (ASE) routes are routes to destinations external to the AS.

Routes imported into OSPF as type 1 routes are supposed to be from EGPs whose external metrics are directly comparable to OSPF metrics. When a routing decision is being made, OSPF will add the internal cost to the AS Border router to the external metric.

Type 2 ASEs are used for EGPs whose metrics are not comparable to OSPF metrics. In this case, only the internal OSPF cost to the AS Border router is used in the routing decision.

From the topology database, each router constructs a tree of the shortest paths with itself as the root. This shortest-path tree gives the route to each destination in the AS. Externally derived routing information appears on the tree as leaves. The link-state advertisement format distinguishes between information acquired from external sources and information acquired from internal routers, so there is no ambiguity about the source or reliability of routes. Externally derived routing information (for example, routes learned from EGP or BGP) is passed transparently through the autonomous system and is kept separate from OSPF's internally derived data. Each external route can also be tagged by the advertising router, enabling a passing of additional information between routers on the borders of the autonomous system.

3.2.1.3 File Transfer Protocol

File Transfer Protocol (FTP) allows hosts to transfer files. FTP provides for such tasks as listing remote directories, changing the current remote directory, creating and removing remote directories, and transferring multiple files in a single request. FTP maintains a secure transport by passing user and account passwords to the foreign host. FTP allows interactive user-oriented sessions.

FTP uses reliable stream transport (TCP/IP) to send the files and uses a TELNET-like connection to transfer commands and replies. FTP also understands several basic file formats, including ASCII, IMAGE, and Local 8. TCP/IP implements FTP in the `ftp` user command and the `ftpd` daemon.

3.2.1.4 Network File System Protocol over UDP transport

The Network File System (NFS) provides access to files via standard UNIX system calls. This allows any program to access files across the network. NFS uses the UDP transport layer; therefore, it has to deal with lost

datagrams. NFS does this by retransmitting requests if a reply has not been received within a reasonable amount of time.

Some requests can be reexecuted on the server without problems, but others (such as file deletion) cause an error if the first request reaches the server but the reply is lost. If the second request is executed, the server finds that the file does not exist and returns an error. NFS servers hold on to such replies and retransmit them if they see a duplicate request.

On the other hand, the protocol is designed so that the servers need no other state information. This allows server performance to be improved by running multiple copies of the server daemon, and also means that server crashes are tolerated with no special code on either client or server.

For more information on NFS, see Chapter 4.

3.2.1.5 Network File System Protocol over TCP transport

Tru64 UNIX contains NFS support over the TCP transport. UDP may still be the preferred transport in local area networks, but for NFS access over wide area, congested, or lossy networks, TCP may offer better performance.

Separate threads are used to maintain some of the performance optimizations made to the UDP code paths. The `nfsiod` daemon spawns kernel threads, instead of forking multiple processes. Each `nfsiod` thread can handle UDP or TCP mounts, so the `nfsiod` command accepts one argument.

For more information on NFS, see Chapter 4.

3.2.1.6 Telnet Protocol

The Telnet Protocol (TELNET) provides a standard method for terminal devices and terminal-oriented processes to interface. TELNET is commonly used by terminal emulation programs that allow you to log in to a remote host. However, TELNET can also be used for terminal-to-terminal communications and interprocess communications.

TCP/IP implements TELNET in the `telnet` user command and the `telnetd` daemon.

3.2.1.7 Trivial File Transfer Protocol

The Trivial File Transfer Protocol (TFTP) can read and write files to and from a foreign host. Like FTP, TFTP can transfer files as either 8-bit NETASCII characters or as 8-bit binary data. Unlike FTP, TFTP cannot be used to list or change directories at a foreign host and it has no provisions

for security, such as password protection. Data usually can be written or retrieved only in public directories.

TCP/IP implements TFTP in the `tftp` user command and in the `tftpd` daemon.

3.2.1.8 Finger Protocol

The Finger Protocol (FINGER) is an application-level Internet protocol that provides an interface between the `finger` command and the `fingerd` daemon. The `fingerd` daemon returns information about the users currently logged into a specified remote host. If you execute the `finger` command specifying a user at a particular host, you obtain specific information about that user. The Finger Protocol must be present at the remote host and at the requesting host. FINGER uses TCP as its underlying protocol.

3.2.1.9 Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) is the standard for mail exchange between machines attached to the Internet. It specifies the format of control messages sent between two machines to exchange electronic mail.

As its name implies, SMTP is simple in design and purpose. Its objective is to provide a reliable and efficient mail delivery system across the links between machines. SMTP does not specify the user interface.

3.2.1.10 Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) is the Internet standard protocol for exchanging network management information. The SNMP agent provides a local or remote network manager with information by accessing Management Information Bases (MIBs). The `snmpd`(8) reference page discusses the configuration, security, cluster support, and supported RFCs for the SNMP agent.

Tru64 UNIX supports both industry standard (IETF RFCs) and Compaq enterprise-specific MIBs. See Section 3.5.5 for more details. Compaq enterprise-specific MIB specifications are located in the `/usr/share/sysman/mibs` directory.

The daemons that provide MIB support are `/usr/sbin/os_mibs`, `/usr/sbin/srvSystem_mib`, `/usr/sbin/svrMgt_mib`, `/usr/sbin/cpq_mibs`, and `/usr/sbin/clu_mibs`.

Tru64 UNIX includes Insight Manager, which uses Web-based access to SNMP MIB data. Section 2.12 discusses Insight Manager.

See Section 3.4.7 for information about the extensible SNMP programming interface.

3.2.1.11 POP3

The operating system supports Version 3 of the Post Office Protocol (POP3) as described in RFC 1939. POP3 is a client/server protocol from Qualcomm, Inc. that allows users to download their email from a mail server to a remote client. After messages are delivered to a server the user connects to the server and downloads the messages to the client machine (a desktop or laptop computer running Windows, MacOS, UNIX, or another operating system). Thereafter, all message processing is local to the client machine and environment. This is the protocol used widely today by Internet Service Providers (ISP) to provide email services for their consumers. For more information, see the *Network Administration* guide.

3.2.1.12 IMAP4

The operating system supports Version 4 of the Internet Message Access Protocol (IMAP4) as described in RFC 2060. IMAP4 is a client/server protocol, based on the Cyrus IMAP4 Revision 1 server from Carnegie Mellon University, that allows mail clients to access mail messages on a server. With it, a user can access his or her mail folders and manipulate the contents remotely without having to log into the server. The protocol allows clients to create, delete, and rename mail folders, to check for new messages and remove old messages, and to retrieve messages selectively for local viewing. In addition, the user can select messages by attributes and parse messages in the RFC 822 and MIME formats. For more information, see the *Network Administration* guide.

3.2.1.13 Resource Reservation Protocol

The Resource ReSerVation Protocol (RSVP) is an Internet network layer defined in RFC 2205. It is one of the components in the management of network bandwidth, and provides a mechanism in which quality-of-service (QoS) requests for specific application data streams or flows, simplex unicast or multicast, can be sent and received through a network. If accepted, these requests reserve a specific amount of network bandwidth for the flow.

Applications can use the RSVP API (RAPI) to request enhanced QoS when the default best effort delivery is unacceptable, for example, for video and audio. The types of QoS that applications may request are defined by Internet Integrated Services (RFC 1633 and RFC 2210).

RSVP on Tru64 UNIX supports on FDDI and Ethernet interfaces and unicast and multicast data flows. For more information, see the *Network Programmer's Guide*.

3.2.2 Transport-Level Protocols

The TCP/IP transport-level protocols (UDP and TCP) allow application programs to communicate with other application programs. The User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP) are the basic transport-level protocols for making connections between Internet hosts. When an application sends a message to the transport layer, UDP and TCP break the information into packets, add a packet header including the destination address, and send the information to the network layer for further processing.

Other protocols and applications use UDP to make datagram connections and TCP to make stream connections. The socket interface implements these protocols.

3.2.2.1 User Datagram Protocol

UDP provides a datagram means of communication between applications on Internet hosts. UDP uses destination protocol ports (abstract destination points within a machine), identified by positive integers, to send messages to one of multiple destinations on a host. The protocol ports receive and hold messages in queues until applications on the receiving host can retrieve them.

UDP relies on the underlying IP to send its datagrams and provides the same connectionless message delivery as IP. It offers no assurance of datagram delivery or duplication protection. However, UDP allows the sender to specify source and destination port numbers for the message and also calculates a checksum of both the data and header. These two features allow the sending and receiving applications to ensure the correct delivery of a message.

3.2.2.2 Transmission Control Protocol

TCP provides reliable stream delivery of data between Internet hosts. Like UDP, TCP uses IP, the underlying protocol, to transport datagrams, and supports the block transmission of a continuous stream of datagrams between process ports. Unlike UDP, TCP provides reliable message delivery and ensures that data is not damaged, lost, duplicated, or delivered out of order to a receiving process. Because of this transport reliability, applications programmers are not required to build communications safeguards into their software.

Both TCP and UDP allow programs to send messages to and receive messages from applications on other hosts, and both use protocol ports on the host to identify the specific destination of the message. The TCP retransmission time-out value is dynamically determined for each connection, based on round-trip time.

TCP has the following operational characteristics:

- Basic Data Transfer

  TCP transfers a continuous stream of 8-bit octets in each direction between its users by packaging some number of bytes into segments for transmission through the Internet system. In general, TCP determines the best time to block and forward packets.

- Reliability

  TCP recovers data that is damaged, lost, duplicated, or delivered out of order by the Internet. To achieve this reliability, TCP assigns a sequence number to each octet it transmits, and requires a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within the time-out interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that are received out of order and to eliminate duplicates. To detect damage, TCP adds a checksum to each segment transmitted, checks it at the receiver, and discards damaged segments.

- Flow Control

  To control how much data is sent, TCP returns the following information with every acknowledgment:

  – The sequence numbers it will accept next; these numbers are always greater than the number of the last segment that was successfully received.

  – The number of octets that the sender is allowed to transmit before receiving further permission.

- Multiplexing

  Many processes within a single host can use TCP communications facilities simultaneously. TCP maintains a set of addresses (ports) within each host. TCP combines the port number with the network address and the host address to uniquely identify each connection endpoint (socket). A pair of sockets uniquely identifies each connection.

- Connections

  TCP must initialize and maintain certain status information for each data stream. The combination of this information, including sockets, sequence numbers, and window sizes, is called a connection. Each connection is unique.

3.2.3 Network-Level Protocols

The Internet network-level protocols (IP, ARP, ICMP) handle machine to machine communications. These protocols provide for transmission and reception of transport requests, and handle network-level control.

3.2.3.1 Internet Protocol

The Internet Protocol (IP) is the primary network-level protocol; it provides unreliable, connectionless packet delivery for the Internet. IP defines the format of all the data sent over the Internet. IP also specifies packet processing and error handling.

IP is connectionless because it treats each packet independently. It is unreliable because it does not guarantee delivery or the order of arrival of packets. However, underlying mechanisms guarantee data integrity, assuming it arrives.

IP provides the interface to the network interface level protocols. The physical connections of a network transfer information in a frame with a header and data. IP uses an Internet datagram, which contains a source host address, along with sequencing and control information.

IP automatically adds an IP header to outgoing packets and removes the IP header from incoming packets before sending them to higher level protocols. IP provides for the universal addressing of hosts in the Internet network.

IP is not a reliable communications facility because it does not require acknowledgments from the sending host, the receiving host, or intermediate hosts.

The total length of IP packets can be configured independently for each interface. Packets are broken up into smaller chunks at gateways and reassembled when they reach their destination.

IP Multicasting

The operating system supports IP Multicasting on a Local Area Network (LAN), as described in RFC 1112, and also supports Version 3.5 of the IP multicast kernel support and Version 3.6 of the `mrouted` implementation of the Distance Vector Multicast Routing Protocol (DVMRP), which provides support for “tunnelling” and “pruning.”

Unlike IP broadcasting, IP multicasting allows packets to be taken off the network only by those clients who have configured their systems to receive

the packets. Packets are accepted or rejected at the hardware level, thereby greatly reducing processing overhead. In addition, IP multicasting does not consume much network bandwidth, because applications do not have to send separate packets with identical data to reach several destinations, as they do with point-to-point connections. With IP multicasting, one packet is sent to all interested hosts.

As a result, IP multicasting is valuable to video conferencing applications and applications that provide constant updates to ever-changing information, like applications that provide stock market quotes.

The IP multicasting code was taken from the public domain, and is supported on all Ethernet and FDDI adapters.

Serial Line IP and Compressed Serial Line IP

The operating system has complete IP support for a serial line, so that users can transfer files or NFS-mount file systems across phone lines. Using the CSLIP `slattach` option, headers can be compressed, thereby increasing performance.

The SLIP/CSLIP code is from OSF/1 Version 1.0. However, because the OSF/1 code did not provide a way to access the CSLIP feature, Compaq modified the `slattach` command to provide the necessary access to CSLIP.

Point-to-Point Protocol

Tru64 UNIX supports the Point-to-Point (PPP) protocol (as defined in RFC 1144, 1171, 1172, 1331, 1332, 1334, 1548, 1549, 1661, and 1662) which provides a method for transmitting datagrams over serial point-to-point links. Unlike SLIP, PPP supports standard encapsulation, simultaneous multiplexing of different network layer protocols, an HDLC frame check sequence for error detection, an HDLC escaping mechanism for use with miscellaneous non-8-bit-transparent telephone and switching equipment, and the dynamic negotiation of IP addresses.

In addition, while SLIP only supports `clist tty` drivers, PPP supports both `clist` and STREAMS-based `tty` drivers, as well as remote logins over LANs.

Note that the PPP code was taken from the public domain and includes contributions identified by the footnoted copyright notices. [1] Certain sections of the PPP code was derived from the RSA Data Security, Inc., MD5 Message-Digest Algorithm.

[1] Copyright (c) 1993 The Australian National University. Copyright (c) 1989 Carnegie Mellon University. Copyright (c) 1991 Gregory M. Christy. Copyright (c) 1989 Regents of the University of California. Copyright (c) 1990 RSA Data Security, Inc.

For more information on PPP, see the *System Administration* guide and the `pppd`(8), `pppstats`(8), and `chat`(8) reference pages.

3.2.3.2 Address Resolution Protocol

The Address Resolution Protocol (ARP) translates Internet addresses into hardware addresses. ARP does not translate addresses for the Serial Line Interface Protocol (SLIP) or Point-to-Point Protocol (PPP) because SLIP and PPP have no hardware address.

ARP dynamically traces Internet addresses to hardware addresses on local area networks. The result of this tracing is called a map. The mapped information is stored in mapping tables. TCP/IP uses ARP to collect and distribute the information for mapping tables.

The kernel maintains the mapping tables, and ARP is not directly available to users or applications. When an application sends an Internet packet to an interface driver, the driver requests the appropriate address mapping. If the mapping is not in the table, an ARP broadcast packet is sent through the requesting interface driver to the hosts on the local area network.

When a host that supports ARP receives an ARP request packet, the host notes the IP and hardware addresses of the requesting system and updates its mapping table, if necessary. If the receiving host's IP address does not match the requested address, the host ignores the request packet. If the IP address does match, the receiving host sends a reply packet to the requesting system. The requesting system stores the new mapping and uses it to transmit future Internet packets.

Unlike most protocols, ARP packets do not have fixed-format headers. Instead, the message is designed to be useful with a variety of network technologies.

3.2.3.3 Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a required part of every IP implementation. ICMP handles error and control messages for IP.

ICMP does the following:

- Tests whether a destination is alive and reachable
- Reports parameter problems with a datagram header
- Performs clock synchronization and transit time estimations
- Obtains Internet addresses and subnet masks
- Provides transport-level reachability information
- Updates routing information

ICMP provides feedback about problems in the communications environment, but does not make IP reliable. That is, ICMP does not guarantee that an IP packet will be delivered reliably or that an ICMP message will be returned to the source host when an IP packet is not delivered or is incorrectly delivered.

ICMP messages are sent in varying situations, including the following:

- When a packet cannot reach its destination
- When a gateway host does not have the buffering capacity to forward a packet
- When a gateway can direct a host to send traffic on a better route

3.3 Supported Transports

Tru64 UNIX supports the following network transports:

- Asynchronous Transfer Mode
- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- Fiber Distributed Data Interface
- Token Ring
- Memory Channel (only available on the TruCluster Server layered product)

3.3.1 Asynchronous Transfer Mode

Tru64 UNIX supports PCI and TURBOchannel machines on 155.5 Mb per second Asynchronous Transfer Mode (ATM) networks; there is also an adapter for the PCI bus that supports 622 Mb ATM networks. ATM is a high-speed, connection-based, cell-switched network that, unlike traditional packet switched networks, can carry different kinds of traffic (voice, video, and data) simultaneously. In addition, ATM provides predictable services to those classes of traffic that require bounded latencies and dedicated bandwidths, and, because ATM separates the physical interface from the datalink layer, the same cell and packet formats can be used over a wide variety of physical interfaces from 1 MB per second to 10 GB per second.

This implementation of ATM consists of permanent virtual circuit support; switched virtual circuit support through ATM Forum UNI 3.0 and 3.1 signalling for point-to-point connections; ATM Forum Integrated Local Management Interface (ILMI) for dynamic network address registration;

Classical IP (as defined by RFC 1577, RFC 1483, and RFC 1626); and LAN Emulation over ATM (as defined by The ATM Forum Version 1 standard). For more information on ATM, see *Asynchronous Transfer Mode* and *Network Administration*.

3.3.2 Ethernet

Tru64 UNIX supports 10 MB per second Ethernet networks.

At the physical and IP levels, the operating system supports an Ethernet network with a Maximum Transfer Unit (MTU) of 1500 bytes at a maximum of 10 MB per second.

In conformance with Ethernet standards, the operating system always ensures a minimum packet size of 60 bytes.

The default MTU at the IP level is 1500 bytes at a maximum of 10 MB per second, although this value can be decreased using the `ifconfig` command.

3.3.3 Fast Ethernet

Tru64 UNIX supports 100 MB per second Fast Ethernet (IEEE 802.3 100Base-TX) network.

MTU sizes at the physical and IP levels are the same as those for regular 10 MB per second Ethernet.

3.3.4 Gigabit Ethernet

Tru64 UNIX supports 1000 MB per second Gigabit Ethernet (IEEE 802.3z 1000Base-T) networks on all PCI-based Alpha hardware platforms.

MTU sizes at the physical and IP levels are the same as those for regular Ethernet, although this value can be changed using the `ifconfig` command.

3.3.5 Fiber Distributed Data Interface

Tru64 UNIX supports 100 MB per second Fiber Distributed Data Interface (FDDI) networks in conformance with RFC 1042 and RFC 1188 on all Alpha hardware platforms.

At the physical level, the operating system supports an FDDI network with a Maximum Transfer Unit (MTU) of 4500 bytes at a maximum of 100 MB per second. At the IP level, the MTU is 4352 bytes at a maximum of 100 MB per second.

The default MTU at the IP level is always 4,352 bytes at a maximum of 100 MB per second, although this value can be decreased using the `ifconfig` command.

3.3.6 Token Ring

Tru64 UNIX supports 4 MB per second and 16 MB per second Token Ring networks and source routing in conformance with RFC 1042.

At the physical level, the operating system supports a Token Ring network with a Maximum Transfer Unit (MTU) of 4472 bytes at a maximum of 4 MB per second and 17800 bytes at a maximum of 16 MB per second. At the IP level, the MTU is 4092 bytes at a maximum of 4 MB per second and 8188 bytes at a maximum of 16 MB per second.

The default MTU at the IP level is always 4092 for both 4 and 16 MB per second, although this value can be increased or decreased using the `ifconfig` command.

3.3.7 Multiple Adapter Support

Tru64 UNIX provides support for a single system to have multiple active network adapters in the same subnet; for example, consider tu0 configured with IP address 192.24.156.20 and tu1 configured with IP address 192.24.156.21, both with the same netmask.

On connection establishment, the kernel chooses the interface that has the fewest number of connections. This connection-balancing effect could lead to greater throughput than on a system with just one network adapter per subnet. See the *Network Administration* manual for information on configuring multiple adapter support.

3.3.8 NetRAIN

The operating system provides support for detecting the physical loss of network connectivity, and subsequent automatic switch over to a working network interface. This feature is called Redundant Array of Independent Network adapters (NetRAIN).

The NetRAIN virtual interface configures two or more interfaces on the same LAN segment into a single interface. One of the physical interfaces is always active while the others remain idle. All interfaces, including the idle interfaces, are constantly monitored to ensure that traffic can flow on each.

If the active interface fails or loses network connectivity, NetRAIN switches network traffic to the next available, working interface. All the context of

the previous interface is maintained (for example, hardware address and multicast addresses). The actual failover time is adjustable depending on your network configuration and operation.

See the *Network Administration* manual, and the `nr`(7) and `ifconfig`(8) reference pages for information on NetRAIN configuration.

3.4 Application Programming Interfaces

The network programming environment includes the programming interfaces for application, kernel, and driver developers writing network applications and implementing network protocols. Additionally, it includes the kernel-level resources that an application requires to process and transmit data, some of which include libraries, data structures, header files, and transport protocols.

The following sections briefly discuss these application programming interfaces that are supported by the operating system:

- X/Open Transport Interface (XTI/TLI)
- BSD Sockets
- System V Release 4.0 STREAMS
- Data Link Interface (DLI)
- Data Link Provider Interface (DLPI)
- Extensible SNMP (eSNMP)
- Basic Sockets API for IPv6
- RSVP API (RAPI)

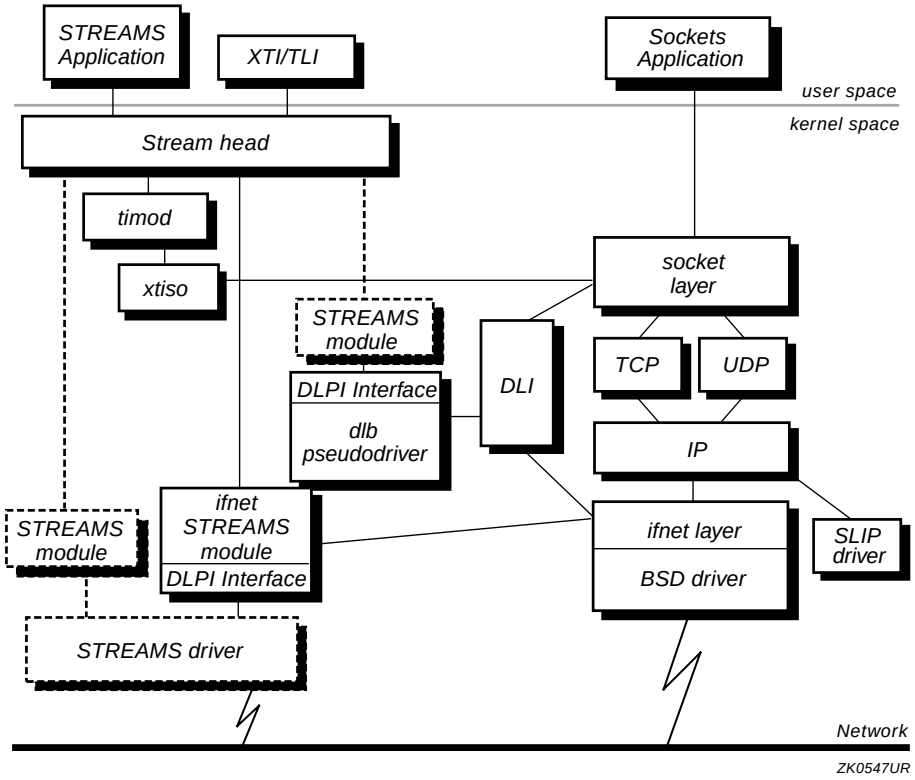
For more detailed information on the network programming environment, see the *Network Programmer's Guide.*

3.4.1 X/Open Transport Interface

The X/Open Transport Interface (XTI) defines a transport layer application interface that is independent of any transport provider. This means that programs written to XTI can be run over a variety of transport providers, such as the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). The application specifies which transport provider to use.

Figure 3–2 illustrates the interaction between XTI and the STREAMS and sockets frameworks.

Figure 3–2: XTI, STREAMS and Sockets Interactions



Depending on the transport provider specified by the application, data can flow along one of two paths:

- If a STREAMS-based transport provider is specified, data follows the same route that it did for an application written to run over STREAMS. It passes first through the Stream head, then to any modules that the application pushed onto the STREAM, and finally to the STREAMS driver, which puts it on to the network. Tru64 UNIX does not provide any STREAMS-based transport providers.
- If a socket-based transport provider (TCP or UDP) is specified, data is passed through `timod` and `xtiso`. The appropriate socket layer routines are called and the data is passed through the internet protocols and `ifnet` layer to the BSD-based driver, which puts it on to the network.

3.4.2 Sockets

Sockets are the industry standard programming interface. In Tru64 UNIX, sockets are the interface to the Internet Protocol suite, for example TCP, UDP, IP, ARP, ICMP, and SLIP. The operating system supports the 4.3BSD

(the default), 4.4BSD, XNS5.0, XNS4.0, and POSIX 1003.1g Draft 6.6 interfaces.

The sockets framework consists of a series of system and library calls, header files, and data structures. Applications can access networking protocols through socket system calls. Applications can also use socket library calls to manipulate network information, for example, the `getservent` library call maps service names to service numbers and the `htonl` library call translates the byte order of incoming data to that appropriate for the local system's architectures.

With sockets, the application in user space passes data to the appropriate socket system calls, which then pass it to the network layer. Finally, the network layer passes it, via the `ifnet` layer, to the BSD driver, which puts it on to the network. For more information, on sockets, see the *Network Programmer's Guide*, X/Open CAE Specification, Networking Services (XNS), Issue 5; *X/Open CAE Specification, Networking Services, Issue 4* (XNS4.0); Protocol Independent Interfaces (POSIX 1003.1g Draft 6.6, Section 5); and the `netintro`(7) reference page.

3.4.3 STREAMS

The STREAMS framework provides an alternative to sockets. The STREAMS interface was developed by AT&T and consists of system calls, kernel routines, and kernel utilities that are used to implement everything from networking protocol suites to device drivers. Applications in user space access the kernel portions of the STREAMS framework using system calls such as `open`, `close`, `putmsg`, `getmsg` and `ioctl`. Tru64 UNIX supports System V Release 4.0 STREAMS from the OSF/1 Version 1.2 code base, which provides support for the STREAMS `tty` interface (although the operating system continues to support the existing CLIST and Berkeley-based `tty` interface). For more information on STREAMS, see the *Network Programmer's Guide.*

3.4.4 Sockets and STREAMS Interaction

Tru64 UNIX provides the `ifnet` STREAMS module to allow programs using BSD-based TCP/IP to access STREAMS-based drivers. It provides the Data Link Bridge (DLB) pseudodriver to allow programs using a STREAMS-based protocol stack to access BSD-based drivers provided on the operating system.

3.4.5 Data Link Interface

The Data Link Interface (DLI) is provided as a backward compatibility feature to ULTRIX operating system. DLI support allows programs written

to DLI on the ULTRIX operating system to access the data link layer. For more information on DLI, see the *Network Programmer's Guide*.

3.4.6 Data Link Provider Interface

The Data Link Provider Interface (DLPI) is a kernel-level interface that maps to the data link layer of the OSI reference model. DLPI frees users from specific knowledge of the characteristics of the data link provider, allowing those characteristics to be implemented independently of a specific communications medium. It is primarily a kernel-level interface targeted for STREAMS protocol modules that use or provide data link services.

Tru64 UNIX supports only a partial subset of the DLPI interface. For more information, see the *Network Programmer's Guide*.

3.4.7 Extensible SNMP Interface

Tru64 UNIX supports extensible SNMP (eSNMP), an application-layer Application Programming Interface (API) that permits user-written programs to function as part of a distributed SNMP agent on a Tru64 UNIX host system.

User programs (subagents) dynamically can register SNMP MIB objects with the eSNMP master agent (`/usr/sbin/snmpd`), and subsequently handle the SNMP protocol operations for those objects.

The distribution of MIB objects between cooperating processes is transparent to SNMP applications, which can access all MIB objects using the standard transport endpoints specified in the SNMP RFCs. The eSNMP API (`libesnmp.so`) uses RFC 2257 (AgentX) for communicating to `snmpd`. This backward compatible change permits subagents to interoperate with any FRC 2257 conforming SNMP agent.

The extensible SNMP development tools are contained in the optional programming subset (PMR). For more information, see *Network Programmer's Guide*.

3.5 Network Administration Software

Tru64 UNIX supports a variety of network administration software, which is briefly described in the following sections.

3.5.1 Networking Commands and Utilities

Tru64 UNIX supports the entire suite of networking commands from OSF Version 1.2, including `finger`, `ftp`, `rdump`, `rdist`, `routed` and `gated`,

telnet and tftp. The bootpd functions have been folded into joind daemon, which provides configurations to clients using either the DHCP or BOOTP protocol.

Additionally, the operating system supports the following Open Network Computing (ONC) Version 4.2 utility programs, which are invoked by the inetd daemon: rwall/rwalld, rusers/rusersd, spray/rsprayd, rup/rstatd, rquotad and pcnfsd.

3.5.2 Ethernet Packet Filter and Packet Filter Applications

The Ethernet packet filter is a software driver interface that demultiplexes networking packet headers, as well as provides reception and transmission of packets containing user-defined network protocols. The packet filter can function also as an Ethernet monitor when used to filter specific network protocols.

Tru64 UNIX supports the following packet filter applications:

- /usr/sbin/rarpd – Reverse ARP daemon

  The reverse ARP daemon responds to RARP requests on a network by sending an IP address to a host that only knows its Ethernet address. It uses the /etc/ethers file to map the Ethernet address to an IP address.

  The reverse ARP daemon can serve IP addresses to remote PC clients. Also, some ULTRIX systems use the Reverse ARP protocol to supply remote stations with their IP address.

- /usr/sbin/tcpdump – TCP/IP tracing and monitoring tool

  Tru64 UNIX supports Version 2.2.1 of the tcpdump utility. This version of tcpdump uses the Berkeley Packet Filter (BPF) language.

  The tcpdump utility is used to debug and analyze TCP/IP network activity, on Ethernet, Token Ring, Memory Channel, FDDI, and ATM networks, and has some support for other protocol suites and interfaces (including PPP, NFS, and loopback). This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.

- /usr/sbin/tcpslice – Log file tool

  The tcpslice utility manipulates tcpdump trace log files by either extracting pieces of or gluing together tcpdump log files. It can select portions of a large tcpdump log file and display selected traces without having to dump the entire log file.

- /usr/sbin/nfswatch – NFS monitoring tool

  Tru64 UNIX supports Version 4.1 of nfswatch from Purdue University. The nfswatch utility is curses-based and displays the NFS traffic between any number of NFS servers and clients on a LAN.

- /usr/sbin/nfslogsum – NFS log file summary tool

  The nfslogsum utility condenses the log files produced by nfswatch into a traffic analysis summary and is very helpful in troubleshooting networks.

Note

The packet filter is an optional kernel subsystem; application programs that make calls to the packet filter kernel routines may fail if the packet filter is not configured in the currently running kernel. For more information, see the packetfilter(7) reference page.

3.5.3 Dynamic Host Configuration Protocol

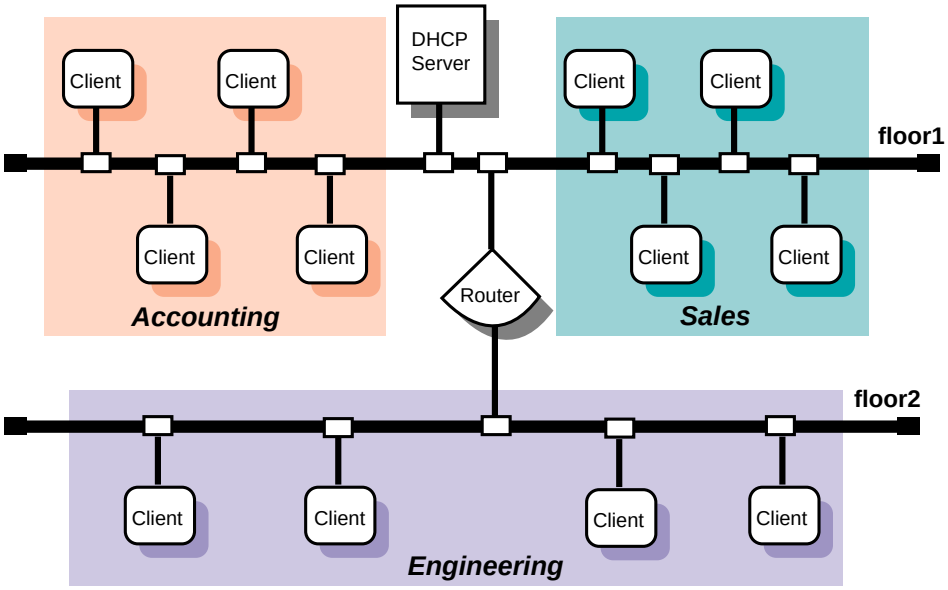
Tru64 UNIX supports the Dynamic Host Configuration Protocol (DHCP) based on JOIN Server Version 4.1 from JOIN Systems, Inc.; it is a client/server framework in which the DHCP server can dynamically assign an IP address to a client as the client boots onto the network. Additionally, a DHCP server can provide configuration information to the client, such as the name of the DNS server or the name of the default router for that client.

For example, when a new system is booted for the first time, the DHCP server assigns that system a unique IP address; if that system is moved to another location on the same LAN (perhaps on a different subnet), the DHCP server ensures that a new IP address appropriate to that subnet is assigned to the system, if necessary, when it boots up for the first time.

As a result, with DHCP customers with hundreds of clients no longer have to worry about the assignment of IP addresses; DHCP assigns IP addresses automatically and requires no intervention by a system administrator.

Figure 3–3 illustrates an example configuration.

Figure 3–3: Dynamic Host Configuration Protocol



ZK-1146U-AI

For more information on DHCP, see the *Network Administration* Guide and the `dhcp`(7) reference page.

3.5.4 The Internet Boot Protocol Daemon

The internet boot protocol daemon, `bootpd`, implements an internet boot protocol (BOOTP) server as defined in RFC 951, RFC 1532, and RFC 1533.

BOOTP is an extensible UDP/IP-based protocol that allows a booting host to configure itself dynamically without having to rely on user intervention. The BOOTP protocol assigns IP addresses to hosts, makes available a file containing a boot program that can be downloaded from a server, provides the address of that server, and the address of an Internet gateway, if one is present.

Like DHCP, the BOOTP protocol supports the centralized management of network addresses.

The `joind` daemon provides the functions of `bootpd`; the `joind` daemon also provides DHCP services.

3.5.5 SNMP Agent

The SNMP agent for this operating system provides access to a great deal of management information typically used in network administration:

- System, network interface, address resolution (ARP), routing (IP and ICMP), and transport layer (TCP and UDP) information is available through the Internet MIB (RFC 1213)
- Host information, such as processes, file systems, memory, attached devices, is available via Host Resources MIB (RFC 1514)
- FDDI interface information is available via the FDDI MIB (RFC 1285)
- Token Ring interface information is available via the Token Ring MIB (RFC 1285)
- Ethernet counters are available via the Ethernet-like Interface MIB (RFC 1398)
- IP routing-related MIBs are available, as described in Section 3.5.6
- Server information such as firmware, memory, and device configuration, and environmental monitoring, is available in the Server System MIB
- Thresholds, alarms, and actions may be configured by the Server Management MIB
- System information is available from Compaq-specific MIBS in support of Insight Manager

The extensible SNMP agent permits the dynamic addition of supported Management Information Bases (MIBs) on any Tru64 UNIX host.

The master agent, API, and base operating system MIB support are all contained in the standard networking subset (CLINET).

The extensible SNMP development tools are contained in the optional programming subset (PGMR).

3.5.6 The gated Daemon

The `gated` daemon allows any host with multiple network interfaces to function as an IP router by participating in various IP routing protocols (for example, RIP, OSPF, EGP, and BGP). The operating system supports the GateD Release 3.5 `gated` daemon from the Merit GateD Consortium, which contains support for the following:

- RIP Version 1 (RFC 1058)

  Stipulates the proper subsuming of host routes, split horizon without poison reverse, and graceful shutdowns.

- RIP Version 2 (RFC 1388)

  Stipulates using IP Multicast where available; supports classless routing; uses next hop (if different).

- OSFP Version 2 (RFC 1247)

  Uses local-wire IP Multicast support, MIB support (RFC 1253), and reconfiguration.

- Support for Routing Table MIB (RFC 1354)
- EGP 2 (RFC 904)

  A complete implementation of the specification, with optimizations for MILNET.

- BGP Versions 2 and 3 (RFC 1163 and RFC 1267)

  Complete implementations of specifications; BGP MIB (RFC 1269); AS path pattern matching RFC 1164); and OSPF/BGP Interaction (RFC 1403).

- BGP Version 4 (RFC 1654)
- DCN HELLO

  Proper subsuming of host routes; split horizon without poison reverse.

- Variable subnet masks through Routing Socket Support and improved synchronization of the kernel routing table
- Routing Table Enhancements

  Based on BSD 4.3 Reno radix tree, `gated` implements filtered routing based on policy. This allows network administrators to control the import and export of routing information by individual protocol, by source and destination autonomous system, source and destination interface, previous hop router, and specific destination address.

  Network administrators can also specify a preference level for each combination of routing information being imported by using a flexible masking capability. Once the preference levels are assigned, `gated` decides which route to use independent of the protocols involved.

- MIB Support for the Following Protocols (“Get Object” Support Only):
  - OSPF V2 MIB (RFC 1253)
  - EGP-MIB (RFC 1213)
  - BGP V3 MIB (RFC 1269)

For more information on `gated`, see the `gated`(8), `gated.conf`(4), `gated.control`(4), `gated.proto`(4), and `gated_intro`(7) reference pages.

3.5.7 The screend Daemon

The `screend` daemon is used in conjunction with the gateway screen facility to decide which IP packets should be forwarded when the system is acting as an IP gateway.

The gateway packet screening facility, on a Tru64 UNIX system acting as a gateway, allows the system manager to control which packets are forwarded or rejected. As a result, the gateway packet screening facility can be used as one part of a comprehensive network security policy.

The facility consists of a kernel-resident mechanism and a user-level daemon, `/usr/sbin/screend`. When a packet is ready to be forwarded, the kernel mechanism submits the packet's headers to the daemon. The `screend` daemon then examines the headers and tells the kernel to forward or reject the packet, based on a set of rules defined in the configuration file, `/etc/screend.conf`.

Optionally, some or all decisions can be logged allowing the network manager to detect improper configurations or potential security problems.

3.5.8 UNIX-to-UNIX Copy Program

The operating system supports the HoneyDanBer version of the UNIX-to-UNIX Copy Program (UUCP), which is a group of programs that supports communications between two computers running UNIX operating systems. The UUCP system enables batched, error-free file transfer and remote command execution between two UNIX systems. The UUCP system is most frequently used to transfer electronic mail, network news, and public domain software over low-speed, low-cost communications links.

The UUCP protocol supports only direct connections between two systems; electronic news and mail delivery depend on third-party forwarding. To facilitate mail and news delivery, most connected sites are willing to relay files for other sites. The UUCP network depends on direct distance dialing networks and off-peak long distance rates for its continued functioning. For more information on UUCP, see the `uucp_setup`(8) reference page.

3.5.9 Local Area Transport

Local Area Transport (LAT) is a protocol that supports communications between host computer systems and terminal servers with terminals, PCs, printers, modems and other devices over local area networks (LANs). LAT software has the features required for a host to function as a service node, so requests for connections can be made by server users. The software also permits host applications to initiate connections to server ports, designated

as application ports, to access remote devices. The LAT driver is STREAMS-based and supports up to 4000 incoming connections, with a theoretical limit of 5000 users.

In Tru64 UNIX, LAT supports both SVR4 and BSD-style terminal devices. Integral serial terminal devices and serial terminal options share the same BSD `tty` namespace as LAT, which means that if special files are allocated for serial lines, those special files will reduce the number of BSD LAT devices that can be configured.

For more information on LAT, see the `lat_intro`(7) reference page and the *System Administration* guide.

3.5.10 Network Interface Monitoring

The `niffconfig` command arranges for one or more network interfaces to be monitored for possible loss of connectivity. Timing parameters that govern how quickly an interface can be declared suspect or dead can be manipulated with this command.

Once an interface has been specified for monitoring, the kernel Traffic Monitor Thread (TMT) checks the connectivity of the monitored interface and, if necessary, informs the Network Interface Failure Finder daemon (`niffd`) to generate traffic for network interface that have been classified inactive. The `niffd` daemon's purpose is to get the interface packet counters to increment, signifying the interface is still alive and well.

See the *Network Administration* manual, and the `niffconfig`(8), `niffd`(8), and `niffmt`(7), reference pages.

3.6 Naming Services

Tru64 UNIX supports the following distributed naming services:

- The Domain Name Service (DNS)
- The Network Information Service (NIS), formerly called Yellow Pages (YP)

The library routines in `/usr/lib/libc.so` allow transparent access to DNS, NIS, and local `/etc` files. The name services configuration file, `/etc/svc.conf`, dictates which naming services are queried, and in what order, for a particular database.

The operating system software allows you to convert from an NIS-distributed environment to a DNS-distributed environment, or to run both services in the same environment. Because the source files for both DNS and NIS can be `/etc` style files, a distributed Berkeley Software

Distribution (BSD) source area can be shared between the two services by means of symbolic links.

3.6.1 Domain Name Service

The Domain Name Service (DNS) is a mechanism for resolving unknown host names and Internet Protocol (IP) addresses that originate from sites on your company's intranet or the Internet.

The implementation of DNS in Tru64 UNIX is based on the Berkeley Internet Name Domain (BIND) service, which is supported by the Internet Software Consortium. BIND service is a client-server model that allows client systems to obtain host names and addresses from DNS servers.

Tru64 UNIX supports BIND Version 8.1.2, which includes the following features:

- Notification and dynamic update of slave servers by master server when DNS resource records are changed (RFC 1996 and RFC 2136)
- New format and designation for default name server configuration file
- Flexible, categorized logging system
- IP-address-based access control for queries, zone transfers, and updates
- Improved performance and security
- Bug fixes

You can use DNS to replace or supplement the host table mapping provided by the local `/etc/host` file or NIS. You should use NIS for all other distributed database applications.

For more information about the DNS environment, DNS planning and configuration, and DNS management, see the *Network Administration* guide, the BIND Configuration File Guide, and the `bind_intro`(7) reference page.

3.6.2 Network Information Service

The Network Information Service (NIS) is a distributed name service that allows participating hosts to share access to a common set of system and network files. NIS allows system administrators to manage these shared files on a single system.

NIS is intended for use in a secure environment only, where gateways do not allow outside access from the Internet to the NIS protocol.

3.7 Time Services

Tru64 UNIX supports the following time services:

- Network Time Protocol (NTP)
- Time Synchronization Protocol (TSP)

Because it can be traced to clocks of high absolute accuracy, NTP provides a more accurate time service than TSP. By contrast, TSP synchronizes time to the average of the network host times. TSP is an acceptable time service if your system is not on the Internet and does not have access to a highly accurate time server; otherwise, NTP is recommended.

3.7.1 Network Time Protocol

The Network Time Protocol (NTP) provides accurate, dependable, and synchronized time for hosts on both wide area networks (like the Internet) and local area networks. In particular, NTP provides synchronization traceable to clocks of high absolute accuracy, and avoids synchronization to clocks keeping bad time.

Tru64 UNIX supports NTP Version 3, based on RFC 1305, which contains the following enhancements to Version 2:

- New algorithms in several clock and peer routines to improve accuracy and stability and reduce errors.
- An authentication mechanism that uses MD5 algorithms for encryption.
- The `ntptrace` utility, which traces a chain of NTP hosts back to their master time source.

Hosts running NTP periodically exchange datagrams querying each other about their current estimate of the time. Using the round-trip time of the packet, a host can estimate the one-way delay to the other. (The delay is assumed to be roughly equal in both directions.) By measuring the one-way delay and examining the timestamps that are returned with the NTP packet, a host computes the difference between its clock time and that of the host it queried.

A host queries a remote host several times over a period and feeds the results from the multiple samples to a digital-filtering algorithm. The algorithm provides a more accurate estimate of the delay, clock offset, and clock stability than could be obtained with a single sample.

NTP messages also contain information about the accuracy and reliability of the time sources. An NTP host connected directly to a highly accurate

time source, such as a radio receiver tuned to a time code signal broadcast by a government agency, is called a stratum 1 server. Every other NTP host adopts a stratum number that is one higher than the host from which it sets its own time. For example, a host synchronized to a stratum 1 server becomes a stratum 2 host. Stratum determination is done automatically, and the stratum of a host can vary as its connectivity changes.

A host running NTP collects information to decide which of the hosts it queries provides the most accurate time. This information includes the output of the digital-filtering algorithm and the stratum numbers of the hosts it queried. By communicating with several other hosts, an NTP host can usually detect those hosts that are keeping bad time, and is able to stay synchronized even if some of the other hosts become unavailable for long periods.

In practice, NTP is able to synchronize clocks to within a few tens of milliseconds even over wide area networks spanning thousands of miles.

For detailed information on NTP, see RFC 1305: *Internet Time Synchronization: the Network Time Protocol*.

3.7.2 Time Synchronization Protocol

The Time Synchronization Protocol (TSP) is the protocol used by the `/usr/sbin/timed` daemon. In its simplest application, the TSP servers on a broadcast network (for example, an Ethernet) periodically broadcast TSP packets. The hosts on the network elect one of the hosts on the network running TSP as a master. The master then controls the further operation of the protocol until it fails and a new master is elected. The master collects time values from the other hosts and computes the average of all the times reported. It then sets its own clock to this average, and tells the other hosts to synchronize their clocks with it.

TSP quickly synchronizes all participating hosts. However, because TSP does not trace time back to sources of known accuracy, it is unable to correct for systematic errors. If a clock drifts significantly, or if a mistake is made in setting the time on a participating host, the average time calculated and distributed by the master can be affected significantly.

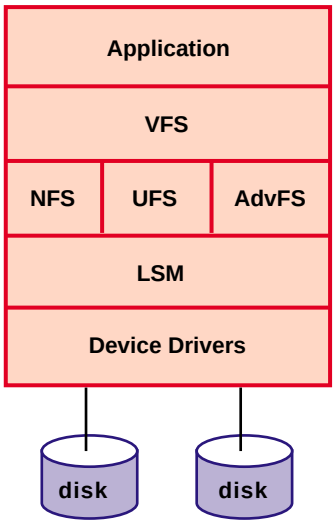
4

File Systems

Tru64 UNIX provides several file systems, which are all accessed through a Virtual File System (VFS) layer, and all of which are integrated with the virtual memory Unified Buffer Cache (UBC).

The file system that you see is handled by the Virtual File System layer, which interacts with the local file system or the networked file system. Under Tru64 UNIX, the default file system is the Advanced File System. From there, the networked file system or the local file system interfaces with the Logical Storage Manager, and in turn, the device drivers and the physical storage devices. Figure 4–1 illustrates this interplay of the file systems, the Logical Storage Manager, and the physical storage devices.

Figure 4–1: File Systems



ZK-1577U-AI

The file systems are:

- Advanced File System (AdvFS)
- UNIX File System (UFS)

- Network File System (NFS)
- Compact Disk Read-Only Memory File System (CDFS)
- Memory File System (MFS)
- The `/proc` File System
- File-on-File Mounting File System (FFM)
- File Descriptor File System

Additionally, the operating system supports storage management for files through the Logical Storage Manager (LSM).

This chapter introduces these software components.

4.1 Virtual File System

The Virtual File System (VFS) is based on the Berkeley 4.3 Reno virtual file system. The VFS presents a uniform interface to users and applications, an interface that is abstracted from the file system layer, in order to allow common access to files, regardless of the file system on which they reside. As a result, file access across different file systems is transparent to the user.

A structure known as a `vnode` contains information about each file in a mounted file system. The `vnodes` are analogous to inodes: they are more or less wrappers around file system-specific nodes. If, for example, a read or write request is made on a file, the `vnode` points the system call to the routine appropriate for that file system. A read request is pointed to `advfs_read` when the request is made on a file in an AdvFS; to `ufs_read` when the request is made on a file in a UFS; or to `nfs_read` when the request is made on a file in an NFS-mounted file system.

The Tru64 UNIX VFS implementation supports Extended File Attributes (XFAs), including support for any application that wants to assign an XFA to a file. Both AdvFS and UFS support XFAs. For more information on XFAs, see the `setproplist`(2) reference page.

4.2 Advanced File System

The Advanced File System (AdvFS) is the default root file system for Tru64 UNIX; it is a log-based file system that provides flexibility, compatibility, data availability, high performance, and simplified system management. AdvFS is designed to handle files and filesets approaching 16 terabytes in length.

The configuration of AdvFS differs from the traditional UNIX file system. In AdvFS, the physical storage layer is managed independently of the

directory layer. System administrators can add and remove storage without unmounting the file system, or halting the operating system. As a result, configuration planning is less complicated and more flexible.

From a user's perspective, AdvFS behaves like any other UNIX file system. You can use the `mkdir` command to create new directories, the `cd` command to change directories, and the `ls` command to list directory contents. AdvFS logical structures, quota controls, and backup capabilities are based on traditional file system design. AdvFS has its own complement of file system maintenance utilities including `mkfdmn` and `mkfset`, which create file systems, and `vdump` and `vrestore`, which are used to back up and restore filesets. AdvFS commands and utilities are described in *AdvFS Administration*.

Without taking an AdvFS system off line, system administrators can perform backups, file system reconfiguration, and file system tuning. End users can retrieve their own unintentionally deleted files from predefined trashcan directories or from clone filesets without assistance from system administrators.

AdvFS supports multivolume file systems, which enables file-level striping (spreading data to more than one volume) to improve file transfer rates for I/O intensive applications. Logical Storage Manager (LSM), which allows volume-level striping, can be incorporated into AdvFS configurations.

AdvFS Utilities, which is licensed separately from the operating system, provides additional file management capabilities and a web-based graphical user interface (GUI) to simplify system administration. The AdvFS GUI, which runs under the Common Desktop Environment (CDE), features menus, graphical displays, and comprehensive online help that make it easy to perform AdvFS operations. In addition, the GUI displays summarized system status information.

The following table lists the main AdvFS features and their benefits.

Feature	Benefit
Rapid crash recovery	Write-ahead logging eliminates the need to use the `fsck` utility to recover from a system failure. The file system recovery time is rapid and independent of file system size.
Extended capacity, scalability	The design supports large-scale storage systems.
High performance	An extent-based file allocation scheme consolidates data transfer.
Disk spanning*	A file or file system can span multiple disks within a shared storage pool.

Feature	Benefit
Unified buffer cache (UBC)	This cache interacts with the virtual memory system to adjust dynamically the amount of physical memory being used to cache file data.
Online defragmentation	System performance improves by defragmentting the data on the disk while the system remains in use. Defragmentation makes file data more contiguous on the storage medium.
Online resizing*	The size of the file system can be dynamically changed by adding or removing disk volumes while the system remains in use. This enables both online storage configuration and online file system maintenance.
File-level striping*	Distributing file data across multiple disk volumes improves file transfer rates.
Online backup*	File system contents can be backed up to media without interrupting the work flow of system users by using fileset clones.
File undelete*	Users can recover deleted files without assistance from system administrators.
Quotas	AdvFS supports quotas for users and groups as well as for filesets.
Graphical user interface*	The GUI simplifies file system management by organizing AdvFS functions into menu-selected tasks and by displaying file system status.

* This feature requires the optional AdvFS Utilities license

The following features were added to AdvFS in Tru64 UNIX Version 5.0:

- Improved disk structure that increases the number of files the file domain can track
- A disk salvage utility that can recover information at the block level from disks that have been damaged
- An improved directory structure that increases the speed of file creation and access
- Enhanced `vdump` and `vrestore` command capability
- Remote device support for backup and restore
- Increased quota limits
- Direct I/O to allow unbuffered, synchronous I/O
- Smooth sync option to promote continuous I/O

- New utilities (such as `vdf`, which displays disk usage for file domains and filesets)
- Metadata display utilities

4.3 Network File System (NFS)

The Network File System (NFS) is a facility for sharing files in a heterogeneous environment of processors, operating systems, and networks. NFS does so by mounting a remote file system or directory on a local system and then reading or writing the files as though they were local.

The Tru64 UNIX environment supports NFS Version 3 and NFS Version 2. The NFS Version 2 code is based on ONC Version 4.2, which is licensed from Sun Microsystems; the NFS Version 3 code is derived from prototype code from Sun Microsystems.

As the operating system supports both NFS Version 3 and Version 2, the NFS client and server bind at mount time using the highest NFS version number they both support. For example, a Tru64 UNIX client will use NFS Version 3 when it is served by an NFS server that supports NFS version 3; however, when it is served by an NFS server running only NFS version 2, the NFS client will use NFS Version 2. For more detailed information on NFS Version 3, see the paper *NFS Version 3: Design and Implementation* (USENIX, 1994).

In addition to the basic NFS services, Tru64 UNIX supports these enhancements:

- NFS over TCP
- Write-gathering
- NFS-locking
- Automounting
- PC-NFS
- WebNFS

4.3.1 NFS Version 3 Features

NFS Version 3 supports all the features of NFS Version 2 as well as the following:

- Improved performance
  - Support for reliable asynchronous writes, which improves write performance over NFS Version 2 by a factor of seven, thereby reducing client response latency and server I/O loading

  - Support for a READDIRPLUS procedure that returns file handles and attributes with directory names to eliminate LOOKUP calls when scanning a directory
  - Support for servers to return metadata on all operations to reduce the number of subsequent GETATTR procedure calls
  - Support for weak cache consistency data to allow a client to manage its caches more effectively
- Improved security
  - Provides an ACCESS procedure that fixes the problems in NFS Version 2 with superuser permission mapping, and allows access checks at file-open time, so that the server can better support Access Control Lists (ACLs)
  - File names and pathnames specified as strings of variable length, with the maximum length negotiated between the client and server using the PATHCONF procedure
- Guaranteed exclusive creation of files

4.3.2 Enhancements to NFS

In addition to the NFS Version 3.0 functionality, the operating system features the following enhancements to NFS:

- NFS over TCP

  NFS has been traditionally run over the UDP protocol. The operating system now supports NFS over the TCP protocol. See the mount(8) reference page.

- Write-gathering

  On an NFS server, multiple synchronous write requests to the same file are combined to reduce the number of actual writes as much as possible. The data portions of successive writes are cached and a single metadata update is done that applies to all the writes. Replies are not sent to the client until all data and associated metadata are written to disk to ensure that write-gathering does not violate the NFS crash recovery design.

  As a result, write-gathering increases write throughput by up to 100 percent and the CPU overhead associated with writes is substantially reduced, further increasing server capacity.

- NFS-locking

  Using the fcntl system call to control access to file regions, NFS-locking allows you to place locks on file records over NFS protecting segments of a shared, NFS-served database. The status

daemon, `rpc.statd`, monitors the NFS servers and maintains the NFS lock if the server goes down. When the NFS server comes back up, a reclaiming process allows the lock to be reattached.

- Automounting

  The `automount` daemon automatically and transparently mounts and unmounts NFS file systems on an as-needed basis. It provides an alternative to using the `/etc/fstab` file for NFS mounting file systems on client machines.

  The `automount` daemon can be started from the `/etc/rc.config` file or from the command line. Once started, it sleeps until a user attempts to access a directory that is associated with an automount map or any directory or file in the directory structure. The daemon awakes and consults the appropriate map and mounts the NFS file system. After a specified period of inactivity on a file system, five minutes by default, the automount daemon unmounts that file system.

  The maps indicate where to find the file system to be mounted and the mount options to use. An individual automount map is either local or served by NIS. A system, however, can use both local and NIS automount maps.

  Automounting NFS-mounted file systems provides the following advantages over static mounts:

  – If NIS maps are used and file systems are moved to other servers, users do not need to do anything to access the moved files. Every time the file systems need to be mounted, the daemon will mount them from the correct locations.

  – In the case of read-only files, if more than one NFS-server is serving a given file system, automount will connect you to the fastest server that responds. If at least one of the servers is available, the mount will not hang.

  – By unmounting NFS-mounted file systems that have not been accessed for more than a certain interval (five minutes by default), the `automount` daemon conserves system resources, particularly memory.

- PC-NFS

  Compaq supports the PC-NFS server daemon, `pcnfsd`, which allows PC clients with PC-NFS configured to do the following:

  – Mount NFS file systems

    The PC-NFS `pcnfsd` daemon, in compliance with Versions 1.0 and 2.0 of the `pcnfsd` protocol, assigns UIDs and GIDs to PC clients so that they can talk to NFS.

The pcnfsd daemon performs UNIX login-like password and user name verification on the server for the PC client. If the authentication succeeds, the pcnfsd daemon then grants the PC client the same permissions accorded to that username. The PC client can mount NFS file systems by talking to the mountd daemon as long as the NFS file systems are exported to the PC client in the /etc/exports file on the server.

Because there is no mechanism in Windows to perform file permission checking, the PC client calls the authentication server to check user's credentials against the file's attributes. This happens when the PC client makes NFS requests to the server for file-access that requires permission checking, such as opening a file.

– Access network printers

The pcnfsd daemon authenticates the PC client and then spools and prints the file on behalf of the client.

- WebNFS

WebNFS is an NFS protocol that allows clients to access files over the Internet in the same way that local files are accessed. WebNFS uses a public file handle that allows it to work across a firewall. This public file handle also reduces the amount of time required to initialize a connection. The public file handle is associated with a single directory (public) on the WebNFS server. See the exports(4), exportfs(2), and nfs_intro(4) reference pages for further information.

4.4 UNIX File System

The UNIX File System (UFS) is a local file system. At one time, UFS was the principal file system, and is still a vital alternative to the use of AdvFS.

UFS is compatible with the Berkeley 4.3 Tahoe release. UFS allows a pathname component to be 255 bytes, with the fully qualified pathname length restriction of 1023 bytes. The Tru64 UNIX implementation of UFS supports file sizes which exceed 2 GB.

UFS supports file block clustering, thereby producing sequential read and write access that is equivalent to the raw device speed of the mass storage device. UFS supports file-on-file mount for STREAMS (see Section 4.8).

4.5 Compact Disk Read-Only Memory File System

The Compact Disk Read-Only Memory File System (CDFS) is a local file system. This implementation supports the ISO-9660 CDFS standard for data interchange between multiple vendors; the High Sierra Group

standard for backward compatibility with earlier CD-ROM formats; and an implementation of the Rock Ridge Interchange Protocol (RRIP), Version 1.0, Revision 1.09. The RRIP extends ISO-9660 system use areas to include: multiple sessions, mixed-case and long file names; symbolic links; device nodes; deep directory structures; user IDs and group IDs and permissions on files; and POSIX time stamps.

All the code for CDFS has been taken from the public domain and enhanced for Tru64 UNIX.

Additionally, the operating system supports the X/Open Preliminary Specification (1991) CD-ROM Support Component (XCDR). The XCDR allows users to examine selected ISO-9660 attributes through defined utilities and shared libraries. The XCDR also allows system administrators to substitute different file protections, owners, and file names for the default CD-ROM files. See the `cdfs`(4) reference page for more information.

4.6 Memory File System

The operating system supports a Memory File System (MFS), which is essentially a UFS that resides in memory. No permanent file structures or data are written to disk, so the contents of an MFS are lost on system reboots, unmountings, or power failures. Because it does not write data to disk, the MFS is a very fast file system, quite useful for storing temporary files or read-only files that are loaded into it after it is created.

For example, if you are building software that would have to be restarted if it failed, the MFS is a good choice to use for storing the temporary files that are created during the build process, because by virtue of its speed, MFS would reduce the build time. For more information about MFS, see the `newfs`(8) reference page.

4.7 The /proc File System

The `/proc` file system is a local file system that enables running processes to be accessed and manipulated as files by the system calls `open`, `close`, `read`, `write`, `lseek`, and `ioctl`.

The `/proc` file system is layered beneath the VFS: it is a pseudo-file system that occupies no actual disk space. You can use the `mount` and `unmount` commands to manually mount and dismount the file system, or you can define an entry for it in the `/etc/fstab` file.

While the `/proc` file system is most useful for debuggers, it enables any process with the correct permissions to control another running process. Thus, a parent/child relationship does not have to exist between a debugger

and the process being debugged. For more information, see the proc(4) reference page.

4.8 File-on-File Mounting File System

The File-on-File Mounting (FFM) file system allows regular, character, or block-special files to be mounted over regular files. FFM is used, for the most part, by the Tru64 UNIX system calls fattach and fdetach of a STREAMS-based pipe (or FIFO). The two system calls are SVR4–compatible. By using FFM, a FIFO, which normally has no file system object associated with it, is given a name in the file system space. As a result, a process that is unrelated to the process that created the FIFO can then access the FIFO.

In addition to programs using FFM through the fattach system call, users can mount one regular file on top of another by using the mount command. Mounting a file on top of another file does not destroy the contents of the covered file; it simply associates the name of the covered file with the mounted file, thereby making the contents of the covered file temporarily unavailable. The covered file can be accessed only after the file mounted on top of it is unmounted. Note that the contents of the covered file are still available to any process that had the file open at the time of the call to fattach or had the file open when a user issued a mount command that covered the file. For more information on FFM, see the ffm(4) reference page.

4.9 File Descriptor File System

The File Descriptor File System (FDFS) allows applications to reference a process's open file descriptors as if they were files in the UFS. The association is accomplished by aliasing a process's open file descriptors to file objects. When the FDFS is mounted, opening or creating a file descriptor file has the same effect as using the dup system call.

The FDFS allows applications that were not written with support for UNIX style I/O to avail themselves of pipes, named pipes, and I/O redirection. The FDFS is not configured into the Tru64 UNIX system; it must be mounted by command or must be placed as an entry in the system's /etc/fstab file. For more information on the FDFS, see the fd(4) reference page.

5

Kernel

The kernel manages the Tru64 UNIX system resources. It can be adjusted for maximum performance by setting system attributes. Furthermore kernel tuning and debugging tools allow the examination of these attributes.

Symmetric multiprocessing and virtual memory are also discussed in this chapter.

5.1 Kernel Tuning

The operating system includes various subsystems that are used to define or extend the kernel. Kernel variables control subsystem behavior or track subsystem statistics since boot time.

Kernel variables are assigned default values at boot time. For certain configurations and workloads, especially memory or network-intensive systems, the default values of some attributes may not be appropriate, so you must modify these values to provide optimal performance.

Although you can use the `dbx` debugger to directly change variable values on a running kernel, Compaq recommends that you use kernel subsystem attributes to access the kernel variables.

Subsystem attributes are managed by the configuration manager server, `cfgmgr`. You can display and modify attributes by using the `sysconfig` and `sysconfigdb` commands and by using the Kernel Tuner, `dxkerneltuner`. In some cases, you can use the `sysconfig` to modify attributes while the system is running. However, these run-time modifications are lost when the system reboots.

For additional information, see the *System Configuration and Tuning* guide.

5.2 System Attributes and Parameters

The following system attributes have been added to support the Advanced File System:

- `AdvfsMinFreeAccess`
- `AdvfsMaxFreeAccessPercent`

- AdvfsDomainPanicLevel
- AdvfsCacheHashSize

The AdvfsMinFreeAccess and AdvfsMaxFreeAccessPercent attributes provide you with more control over the number of available AdvFS access structures on your system. Access structures are the in-memory representation of an AdvFS file. These attributes supersede the AdvfsAccessCleanupPercent attribute, which has been removed.

The AdvfsDomainPanicLevel attribute controls what happens during an AdvFS domain panic. By default, when an AdvFS domain panic occurs, this attribute enables the system to create a crash dump without crashing the system.

The AdvfsCacheHashSize attribute allows you to adjust the size of AdvFS buffer cache hash table size. AdvFS configures its buffer cache hash table size based on the AdvfsCacheHashSize value.

Additionally, the default value of the AdvfsAccessMaxPercent attribute has changed.

For more information, see the *AdvFS Administration* and *System Configuration and Tuning* guides and the sys_attrs_advfs(5) reference page.

5.3 Enhanced Kernel Debugging

The dbx debugger supports a read-only examination of a locally running kernel, as well as the debugging of kernel core files through the use of the -k switch.

The following two features have been added to dbx:

- A -remote switch to enable the remote, breakpoint debugging of a running kernel across a serial line

  The protocol is multibyte, and caching as well as a multithread extension are supported.

- A front-end to dbx, called kdbx, which supports not only the entire suite of dbx commands, but a C library API that allows programmers to write C programs to extract and format kernel data more easily than they can with just dbx -k or dbx -remote.

  The kdbx front-end ships with several ready-made extensions in the file /usr/var/kdbx.

For more information on kernel debugging, see the guide *Kernel Debugging*.

5.4 Symmetric Multiprocessing

Symmetrical multiprocessing (SMP) is the ability of two or more processes (or multiple threads of a threaded application) to execute simultaneously on two or more CPUs. This concurrency of execution greatly improves performance. Additionally, it affords you the opportunity to extend the life and increase the cost-effectiveness of their multiprocessor systems by adding CPU cards (and their compute power) to their multiprocessors rather than buying more systems.

Tru64 UNIX supports an implementation of SMP that is designed to optimize the performance of **compute servers** (systems dedicated to compute-bound, multithreaded applications) and **data servers** (file servers, DBMS servers, TP systems, and mail routers that serve a large number of network clients). In addition, Tru64 UNIX supports multithreaded application development in an SMP environment. Note that SMP does not adversely affect using a multiprocessor as a timesharing system.

The Tru64 UNIX SMP implementation makes use of simple locks (also called spin locks, because they "spin" for a specified period of time waiting for held locks to be freed before timing out), complex locks (read/write locks that can block waiting for a lock to be freed), and in very rare cases where locks would not be of benefit, funneling, whereby a process is forced to execute on a specified CPU.

The Tru64 UNIX SMP implementation also endeavors to achieve as much concurrency as possible by reducing the size of the system state that must be protected by locks, thereby reducing the necessity for locks and their attendant overhead.

The operating system, including the kernel, is fully parallelized so that multiple processes or multiple threads can run simultaneously on multiple CPUs. The operating system uses concurrency and its locking strategy to ensure the integrity of the same kernel data structures. Multiple processes and multiple threads may access the same kernel data structures, but the operating system maintains that this access is performed in a logical order. Multiple processes and multiple threads may not hold and request each others' locks, deadlocking the system. There are no architectural limits on the number of CPUs supported.

Tru64 UNIX SMP also supports processor affinity, the ability to bind a particular process to a specified CPU, and load balancing, whereby the scheduler attempts to distribute all runable processes across all available CPUs. (Note that load balancing will not override processor affinity).

To improve performance, the scheduler also attempts to execute each process on the last CPU where it ran to take advantage of any state that may be left in that CPU's cache.

SMP is configurable and any of the following five modes can be configured at system boot time:

- Uniprocessing
- Optimized real-time preemption
- Optimized SMP
- Optimized real-time preemption and SMP
- Lock debug mode

When uniprocessing is set, only those locks necessary to support multiple threads are compiled into the kernel at system boot time.

When lock debug mode is set, the system:

- Checks the lock hierarchy and minimum system priority level (SPL)
- Stores debugging information by classes and maintains lock statistics
- Records the simple locks that are held by each CPU in CPU-specific arrays
- Records all of the complex locks that a thread is holding in the thread structure

This debugging information can be accessed with the `dbx` debugger.

In addition, the development environment supports multithreaded application development. The `dbx`, `profile`, and `pixie` utilities support multiple threads, and the system includes thread-safe libraries.

For information on the Tru64 UNIX development environment and the threads package that Tru64 UNIX supports, see the *Programmer's Guide* and the *Guide to DECthreads*. For information on configuring SMP, see the *System Administration* guide and the *System Configuration and Tuning* guide.

5.5 Virtual Memory

The virtual memory subsystem performs the following functions:

- Allocates memory to processes
- Tracks and manages all the pages in the system
- Uses paging and swapping to ensure that there is enough memory for processes to run and to cache file system I/O

The total amount of physical memory is determined by the capacity of the memory boards installed in your system. The system distributes this memory in 8 KB units called pages. The system distributes pages of physical memory among three areas:

Wired memory	Memory is wired statically at boot time and dynamically at run time. At boot time, the operating system and the Privileged Architecture Library (PAL) code wire a contiguous portion of physical memory in order to perform basic system operations. Static wired memory is reserved for operating system data and text, system tables, the metadata buffer cache, which temporarily holds recently accessed UNIX File System (UFS) and CD-ROM File System (CDFS) metadata, and the Advanced File System (AdvFS) buffer cache. Static wired memory cannot be reclaimed through paging.
Virtual memory	The virtual memory subsystem uses a portion of physical memory to cache processes' most recently accessed anonymous memory (modifiable virtual address space) and file backed memory. The subsystem allocates memory to competing processes and tracks the distribution of all the physical pages. This memory can be reclaimed through paging and swapping.
Unified Buffer Cache	The Unified Buffer Cache (UBC) uses a portion of physical memory to cache most recently accessed file system data. The UBC contains actual file data for reads and writes and for page faults from mapped file regions and also AdvFS metadata. By functioning as a layer between the operating system and the storage subsystem, the UBC can decrease the number of disk operations. This memory can be reclaimed through paging.

The virtual memory subsystem and the UBC compete for the physical pages that are not wired. Pages are allocated to processes and to the UBC, as needed. When the demand for memory increases, the oldest (least recently used) pages are reclaimed from the virtual memory subsystem and the UBC, moved to swap space, and then reused. Various attributes control

the amount of memory available to the virtual memory subsystem and the UBC and the rate of page reclamation.

5.5.1 Managing and Tracking Pages

The virtual memory subsystem allocates physical pages to processes and the UBC, as needed. Because physical memory is limited, these pages must be periodically reclaimed so that they can be reused.

The virtual memory subsystem uses page lists to track the location and age of all the physical memory pages. At any one time, each physical page can be found on one of the following lists:

Wired list	Pages that are wired and cannot be reclaimed
Free list	Pages that are clean and are not being used (the size of this list controls when page reclamation occurs)
Active list	Pages that are being used by the virtual memory subsystem or the UBC To determine which pages should be reclaimed first, the page stealer daemon identifies the oldest pages on the active list and designates these least recently used pages as follows: • Inactive pages are the oldest pages that are being used by the virtual memory subsystem. • UBC least recently used pages are the oldest pages that are being used by the UBC.

5.5.2 Prewriting Modified Pages

The virtual memory subsystem attempts to keep memory pages clean to ease the recovery of memory shortages. When the virtual memory subsystem anticipates that the pages on the free list will soon be depleted, it prewrites to swap space the oldest modified (dirty) inactive pages. In addition, when the number of modified UBC least recently used pages exceeds 10 percent of the total UBC least recently used pages, the virtual memory subsystem prewrites to swap space the oldest modified UBC least recently used pages.

5.5.3 Using Attributes to Control Paging and Swapping

When the demand for memory significantly depletes the free list, paging begins. The virtual memory subsystem takes the oldest inactive and UBC least recently used pages, moves the contents of the modified pages to swap space, and puts the clean pages on the free list, where they can be reused.

If the free page list cannot be replenished by reclaiming individual pages, swapping begins. Swapping temporarily suspends processes and moves entire resident sets to swap space, which frees large amounts of physical memory.

The point at which paging and swapping start and stop depends on the values of various tunable virtual memory subsystem kernel attributes.

Because the UBC competes with the virtual memory subsystem for the physical pages that are not wired by the kernel, the allocation of memory to the UBC can affect file system performance and paging and swapping activity. The UBC is dynamic and consumes varying amounts of memory in order to respond to changing file system demands.

By default, the UBC can consume up to 100 percent of memory. However, part of the memory allocated to the UBC is only borrowed from the virtual memory subsystem. When paging starts, borrowed UBC pages are the first to be reclaimed. The amount of memory allocated to the UBC can be controlled by various virtual memory subsystem kernel attributes.

5.5.4 Paging Operation

When the memory demand is high and the number of pages on the free page list falls below the paging threshold, the virtual memory subsystem uses paging to replenish the free page list. The page reclamation code controls paging and swapping. The page out daemon and task swapper daemon are extensions of the page reclamation code.

The page reclamation code activates the page stealer daemon, which first reclaims the pages that the UBC has borrowed from the virtual memory subsystem, until the size of the UBC reaches the borrowing threshold (the default is 20 percent). If the reclaimed pages are dirty (modified), their contents must be written to disk before the pages can be moved to the free page list. Freeing borrowed UBC pages is a fast way to reclaim pages, because UBC pages are usually unmodified.

If freeing UBC borrowed memory does not sufficiently replenish the free list, a page out occurs. The page stealer daemon reclaims the oldest inactive and UBC least recently used pages.

Paging becomes increasingly aggressive if the number of free pages continues to decrease. If the number of pages on the free page list falls below 20 pages (by default), a page must be reclaimed for each page taken from the list. To prevent deadlocks, if the number of pages on the free page list falls below 10 pages (by default), only privileged tasks can get memory until the free page list is replenished. Both these limits are controlled by tunable attributes

Page out stops when the number of pages on the free list rises above the paging threshold. If paging individual pages does not sufficiently replenish the free list, swapping is used to free a large amount of memory.

5.5.5 Swapping Operation

If there is a high demand for memory, the virtual memory subsystem may be unable to replenish the free list by reclaiming pages. Swapping reduces the demand for physical memory by suspending processes, which dramatically increases the number of pages on the free list. To swap out a process, the task swapper suspends the process, writes its resident set to swap space, and moves the clean pages to the free list. Swapping can have a serious impact on system performance.

Idle task swapping begins when the number of pages on the free list falls below the swapping threshold (the default is 74 pages) for a period of time. The task swapper then suspends all tasks that have been idle for 30 seconds or more.

If the number of pages on the free list continues to decrease, hard swapping begins. The task swapper suspends, one at a time, the tasks with the lowest priority and the largest resident set size.

Swapping of an individual task stops when the number of pages on the free list reaches the high water swapping threshold (the default is 1280).

A swap in occurs when the number of pages on the free list has been sufficiently replenished for a period of time. The task's working set is paged in from swap space and it can now execute. By default, a task must remain in the swapped in state for one second before it can be swapped out.

Increasing the rate of swapping (swapping earlier during page reclamation) increases throughput. As more processes are swapped out, fewer processes are actually executing and more work is done. Although increasing the rate of swapping moves long sleeping threads out of memory and frees memory, it degrades interactive response time. When an swapped out process is needed, it will have a long latency.

Decreasing the rate of swapping (swap later during page reclamation), improves interactive response time, but at the cost of throughput.

5.5.6 Choosing a Swap Space Allocation Mode

There are two modes that you can use to allocate swap space. The modes differ in how the virtual memory subsystem reserves swap space for anonymous memory (modifiable virtual address space). Anonymous memory is memory that is not backed by a file, but is backed by swap space (for example, stack space, heap space, and memory allocated by the `malloc` function). There is no performance benefit attached to either mode:

Immediate mode	This mode reserves swap space when a process first allocates anonymous memory. Immediate mode is the default swap space allocation mode and is also called eager mode. This mode may cause the system to reserve an unnecessarily large amount of swap space for processes. However, it ensures that swap space will be available to processes if it is needed.
Deferred mode	This mode reserves swap space only if the virtual memory subsystem needs to write a modified virtual page to swap space. It postpones the reservation of swap space for anonymous memory until it is actually needed. Deferred mode is also called lazy mode. This mode requires less swap space than immediate mode and may cause the system to run faster because it requires less swap space bookkeeping. However, because deferred mode does not reserve swap space in advance, the swap space may not be available when a process needs it, and processes may be killed asynchronously.

In addition, you can override the system wide swap space allocation mode for a specific command or application by using the `swapon` command; see the `swapon`(8) reference page for further information.

5.5.7 Using Swap Buffers

To facilitate the movement of data between memory and disk, the virtual memory subsystem uses synchronous and asynchronous swap buffers. The virtual memory subsystem uses these two types of buffers to immediately

satisfy a page in request without having to wait for the completion of a page out request, which is a relatively slow process.

Synchronous swap buffers are used for page in page faults and for swap outs. Asynchronous swap buffers are used for asynchronous page outs and for prewriting modified pages.

5.5.8 Unified Buffer Cache

The Unified Buffer Cache (UBC) is a Tru64 UNIX virtual memory feature. The UBC uses a portion of the machine's physical memory to cache the most recently accessed file system data. The UBC contains actual file data, which includes reads and writes from conventional file activity, page faults from mapped file sections, and AdvFS metadata.

The UBC shares (contends for) physical memory pages with the virtual memory subsystem, but not pages that are wired by the kernel. The UBC is dynamic, consuming varying amounts of memory in order to respond to changes in file system demands for its service. For information about the UBC, see *System Configuration and Tuning*.

5.6 Device Support

The operating system supports a number of various devices; for specific devices, see the *Software Product Description*.

The kernel now supports hot-swap I/O devices. This provides the capability to automatically fault-in a device driver when a hot-swappable I/O device comes on line.

When the hardware code detects a new device is hot-plugged in and determines that the device driver is not present in the kernel, it can make a single kernel function call to automatically load the device's driver into the kernel. Additionally, hot-swapping provides the flexibility of not having to prebuild the kernel subsystem or driver into the kernel. Instead, it can be faulted in when the device is first accessed.

6

Development Environment

This chapter highlights the major features provided by the development environment for Tru64 UNIX.

The development environment provided by Tru64 UNIX includes the following features:

- A standards compliant, fully optimized C compiler that produces extremely efficient code to exploit fully the 64-bit address space of the Alpha architecture
- Debuggers that support C, Assembler, Fortran (f77 and f90), C++, Pascal, and Ada
- Support for shared libraries, threads, and versioning
- Run-time libraries
- A Java™ development kit (JDK)

In addition, Tru64 UNIX supports internationalization, standard UNIX development tools (for example, `awk`, `lint`, `make`, and `prof`), and run-time libraries for Ada, C++, Cobol, Fortran, and Pascal programs.

For more detailed information, see the *Programmer's Guide*, the guide *Programming Support Tools*, *Assembly Language Programmer's Guide*, and *Writing Software for the International Market*.

6.1 System C Compiler

The Compaq C compiler for Tru64 UNIX is a standards compliant, full-featured, highly-optimizing compiler that was specifically developed to exploit the Alpha architecture. In particular, the system C compiler recognizes these popular C dialects:

- ANSI and ISO C with extensions (`-std`, the default)
- Traditional K&R C (`-std0`)
- Strict ANSI and ISO C (`-std1`, `-isoc94`)
- Microsoft C (`-ms`)
- VAX C (`-vaxc`)

The system C compiler complies with the following standards:

- ANSI and ISO C (`cc -std1`)
- XPG4-UNIX (`c89 -D_XOPEN_SOURCE_EXTENDED`)

For more information on standards compliance, see the `standards`(5) reference page.

The system C compiler supports language extensions including:

- OpenMP parallel decomposition directives (`-omp`, `-check_omp`, `-mp`)
- Microsoft C structured exception handling (`try-except` and `try-finally`) and thread-local storage
- 32-bit pointers (64-bit pointers are the default) to reduce memory consumption and facilitate porting (`-taso`, `-xtaso`, `-xtaso_short`; see the `protect_headers_setup`(8) reference page)
- User-defined assembly language sequences using `#pragmas` and intrinsic functions that expand and optimize these instruction sequences inline
- `#pragmas` and command line options for controlling data alignment (`-assume`, `-misalign`, `-Zp`, `-member_alignment`)

The system C compiler supports enhancements for mathematical computing such as:

- IEEE floating point (including proper handling for exceptional conditions like NaN, INF, and so forth)
- Fast math mode (INF, NaN, and so forth, translated to avoid exception handling)
- Quad-precision (128-bit) floating point representation for `long double`)

For more information on the Compaq C compiler for Tru64 UNIX, see the `cc`(1) reference page.

6.2 Debuggers

Tru64 UNIX supports the following source code debuggers:

- `dbx`
- `ladebug`

6.2.1 The dbx Debugger

The `dbx` debugger can be used to debug programs at the source-code level or the machine-code level. The interface to `dbx` is a command line interface. Features supported by dbx include:

- Debugging programs written in C, Fortran (f77 and f90), assembly language, or Pascal
- Debugging active kernels, core dumps, programs that use multiple threads (by using `/proc` to attach to running processes), and shared libraries
- Analyzing kernel core dumps
- Patching the on-disk copy of either user programs or the kernel
- Performing multiprocess debugging and debugging across `fork` and `exec` calls

6.2.2 The ladebug Debugger

The `ladebug` debugger is a source level, object-oriented symbolic debugger that has both a graphical user interface (GUI) and a command line interface similar to the `dbx` command line interface.

The `ladebug` debugger provides the following functions:

- Attaching to and detaching from running processes
- Detecting and debugging across `fork` and `exec`
- Debugging multiple processes
- Debugging multithreaded programs, either DECthreads applications or kernel modules that make use of kernel threads
- Debugging programs written in C++, C, Fortran 77, Fortran 90, Ada, Cobol, and Assembler

  Note that `ladebug` is a full C++ debugger which demangles C++ names, understands C++ expressions, provides support for inline functions, templates, and C++ exceptions.

  Also note that the support for f77 and f90 includes case insensitivity, common blocks, alternate entry points, language-dependent type printing, and assume shape arrays.
- Debugging machine level code
- Debugging running programs or core dumps
- Debugging shared objects
- Catching unaligned access problems
- Debugging active kernels, either locally or remotely, and analyzing kernel crash dumps
- Evaluating expressions using the syntax of the source programming language

- Remote debugging of programs running on different target machines by way of the remote debugging server.

  Note that the `ladebug` remote debugging protocol is also available along with the C source code for a sample remote debugging server that adheres to the protocol.

- Internationalization

  Note that internationalization support is available in a separate kit. The internationalized `ladebug` debugger accepts multibyte characters as input, and outputs local language characters according to the current global locale set in the debugger. It also supports the `wchar_t` datatype in C and C++.

6.3 Profiling Tools

Tru64 UNIX provides the following profiling toolkit:

- ATOM

  Provides a flexible code instrumentation interface that is capable of building a wide variety of user-defined program analysis tools. The toolkit comprises an instrumentation control tool and a library whose procedural interface enables you to develop special-purpose instrumentation and analysis tools easily.

  Tru64 UNIX provides the following ATOM-based instrumentation and analysis tools:

  – `hiprof`

    A call-graph profiling tool with output that can be postprocessed by `gprof`.

  – `third` (Third degree)

    A tool to find memory leaks and check for incorrect memory accesses.

  – `pixie`

    A superset of the existing `pixie` basic block profiler, which can profile a program's executables and its shared libraries. The output of `pixie` can be analyzed by `prof`.

Tru64 UNIX also supports the following profiling tools:

- `gprof`

  For programs compiled with the `-pg` option, it displays how many calls that named procedures have made to each other and how much CPU time each procedure consumed. It does this by using PC sampling statistics. It also analyzes the output of programs instrumented with `hiprof`.

- `prof`

  For programs compiled with the `-p` option, it displays how much CPU time was consumed by each procedure in a program and its shared libraries. It does this using PC-sampling statistics. It also analyzes the output of programs instrumented with `pixie` or monitored with `uprofile` and `kprofile`.

- `uprofile` and `kprofile`

  These use the Alpha chip's built-in performance counters to sample a variety of events in the CPU during the execution of the kernel or an application program. These tools report on CPU cycles, memory and cache effects, and so forth, which `prof` can then analyze.

For more information on profiling tools, see the *Programmer's Guide* and the appropriate reference pages.

6.4 Shared Libraries

Tru64 UNIX provides a full complement of dynamic shared libraries, compatible with System V semantics for shared library loading and symbol resolution as well as the System V API for dynamic loading (`dlopen`, `dlclose`, `dlsym`, and `dlerror`). Because they allow programs to include only information about how to load and access routines rather than the routines themselves, shared libraries increase system performance, reduce disk and memory requirements, and simplify system management.

Table 6–1 lists shared libraries that Tru64 UNIX supports.

Table 6–1: Tru64 UNIX Shared Libraries, /usr/shlib

Library /usr/shlib	Description
`libDXm.so`	Motif Extensions library
`libDXterm.so`	Support routines for dxterm terminal emulator
`libDtHelp.so`	CDE online help routines
`libDtMail.so`	Shared library support for the `dtmail` CDE mail utility
`libDtSvc.so`	CDE service routines for desktop management
`libDtTerm.so`	Shared library support for the CDE `dtterm` terminal emulator utility
`libDtWidget.so`	Shared library of CDE widgets to supplement Motif widget
`libICE.so`	Inter-Client Exchange library, which enables the building of protocols
`libMrm.so`	Motif Resource Manager library

Table 6–1: Tru64 UNIX Shared Libraries, /usr/shlib (cont.)

Library /usr/shlib	Description
libSM.so	The X Session Management Protocol (XSMP) provides a uniform mechanism for users to save and restore their sessions using the services of a network-based session manager. It is built on ICE and is the C interface to the protocol
libUil.so	The callable Motif UIL (User Interface Language) compiler used by applications that want to compile UIL at run time.
libX11.so	Xlib library
libXETrap.so	Xtrap extension client side library
libXIE.so	X Imaging extension client-side run-time library (V5)
libXaw.so	X Athena widgets run-time library
libXext.so	X client-side extension library
libXi.so	X Input extension client-side library
libXie.so	X Imaging extension client-side run-time library (V3)
libXm.so	Motif widgets library
libXmu.so	X miscellaneous utilities run-time library
libXp.so	Xprint extension client side run-time library
libXt.so	X Intrinsics library - X Toolkit
libXtst.so	A library of routines for X clients to make use of the XTEST extension
libXv.so	X video extension client-side run-time library
libaio.so	POSIX real-time, asynchronous I/O functions
libaio_raw.so	POSIX real-time, asynchronous I/O functions (raw disk and tape only)
libaud.so	Security auditing library
libbkr.so	Motif help system library
libc.so	C library
libc_r.so	Threadsafe libc (link to libc.so, not for new applications)
libcdrom.so	Rock Ridge Extensions to CDFS library
libchf.so	CDA/Imaging signal-handling routines
libcmalib.so	CMA threads library
libcurses.so	Curses screen control library

Table 6–1: Tru64 UNIX Shared Libraries, /usr/shlib (cont.)

Library /usr/shlib	Description
libcxx.so	C++ run time library
libdb.so	Database routines
libdnet_stub.so	DECnet library
libesnmp.so	Extensible SNMP library
libexc.so	Library that provides support for exception handling.
libfilsys.so	File system utility library
libiconv.so	Internationalization codeset conversion routines
libjava.so	Java library
libjpeg.so	JPEG library
libm.so	Portable mathematics library
libmach.so	Mach library
libmath.so	Math library
libmmedia.so	Multimedia library
libmxr.so	Library used by mxr, the ULTRIX binary interpreter for OSF/1
libmsfs.so	AdvFS system call interface library
libndb.so	Database routines
libots.so	Compiler run-time support
libots3.so	Compiler run-time support
libpacl.so	POSIX Access Control List library
libproplist.so	VFS Extended File Attributes library
libpset.so	Processor set routines
libpthread.so	Application Programming Interface for Tru64 UNIX threads
libpthreads.so	DECthreads library
libsecurity.so	Enhanced security library
libsm_x.so	Systems management graphical support library; no user level interfaces available
libtcl.so	Base Tool Command Language (TCL) support library
libtclx.so	Extended TCL support library
libtk.so	Graphical TCL (TK) extensions library

Table 6–1: Tru64 UNIX Shared Libraries, /usr/shlib (cont.)

Library /usr/shlib	Description
libtkx.so	Graphical extended TCL support library
libtli.so	XTI library
libtt.so	SunSoft ToolTalk routines
libxnet.so	X/Open Interfaces library
libxti.so	XTI library

The Tru64 UNIX Version 5.0 implementation of X11R6.3 and Motif makes use of both static and shared libraries. Shared libraries allow client programs to make use of the latest library code without recompiling, and also saves memory and disk space.

Table 6–2 lists the shared X11 libraries that Tru64 UNIX supports.

Table 6–2: Tru64 UNIX Shared Libraries, /usr/shlib/X11

Library /usr/shlib/X11	Description
libPcl.so	Support for PCL printer language for the X print server
libPs.so	Support for PostScript for the X print server
libXau.so	X Authorization library
libXdmDecGreet.so	Motif loadable greeter library
libXdmGreet.so	Athena-style loadable greeter library
libXdmcp.so	X Display Manager control program library
lib_dec_ati32.so	Support for the ATI Mach32 graphics adapter
lib_dec_ati64.so	Support for the ATI Mach64 graphics adapter
lib_dec_cirrus.so	Device support for the Cirrus VGA graphics card
lib_dec_comet.so	Device support for the ELSA Gloria Synergy graphics adapter
lib_dec_ffb.so	Supports the ZLX-E, ZLXp-E, and ZLXp2-E graphics accelerators for 2D drawing operations
lib_dec_ffb_ev5.so	Support for the ZLX-E, ZLXp-E, and ZLXp2-E graphics accelerators for 2D drawing operations on EV5 systems

Table 6–2: Tru64 UNIX Shared Libraries, /usr/shlib/X11 (cont.)

Library /usr/shlib/X11	Description
lib_dec_gen.so	Support for unrecognized VGA class graphics adapters
lib_dec_gen_linear.so	Support for unrecognized VGA class graphics adapters for EV6 systems
lib_dec_s3.so	Support for S3 Trio graphics adapter
lib_dec_s3_linear.so	Support for S3 Trio graphics adapter for EV6 systems
lib_dec_sfb.so	Support for the SFB (HX) Turbochannel graphics adapter
lib_dec_smt.so	Shared memory transport library
lib_dec_triton.so	Support for the Qvision graphics adapter
lib_dec_tx.so	Support for the TX Turbochannel graphics adapter
lib_dec_vga.so	Support routines for use by all VGA class graphics adapters
lib_dec_wd.so	Support for the Western Digital graphics adapter
lib_dec_ws_prt.so	Low-layer operating system interface for the X print server
lib_dec_ws.so	Low-layer operating system interface for the X server
lib_dec_xi_db3.so	Support library for the Drawing Board III graphics tablet
lib_dec_xi_pcm.so	Dynamically-loadable X input extension library that supports the dial and box
lib_dec_xi _serial_mouse.so	Support library for the serial mouse
lib_dec_xv_tx.so	X video extension support for the TX graphic option
libcfb.so	Color frame buffer library
libcfb16.so	16-bit visual support for the color frame buffer
libcfb32.so	32-bit visual support for the color frame buffer
libdbe.so	Double buffer extension library
libdix.so	Device-independent portion of the X Server
libdixie.so	With libmixie.so, supports the X Image Extensions (XIE) Extension library

Table 6–2: Tru64 UNIX Shared Libraries, /usr/shlib/X11 (cont.)

Library /usr/shlib/X11	Description
libextAppgroup.so	Support for the application group extension
libextMITMisc.so	MIT-sundry-nonstandard extension library
libextMultibuf.so	Multibuffering extension library
libextScrnSvr.so	MIT-screen-saver extension library
libextSecurity.so	Support for the X Security extension
libextSync.so	SYNC extension library
libextXCMisc.so	XC-MISC extension library
libextXp.so	Support for the X print extension
libextbigreq.so	Big requests extension library
libextdpms.so	Support for the Display Power Management (DPMS) extension
libextkme.so	Keyboard-management-extension
libextshape.so	Shape extension library
libextshm.so	MIT-SHM extension library
libextxtest.so	XTEST extension library
libextxtrap.so	DEC-XTRAP extension library
libfont.so	Font access library
libfr_Speedo.so	Loadable font renderer library
libfr_Type1.so	Loadable font renderer library
libfr_fs.so	Loadable X Server font renderer for using a font server
liblbx.so	Support for the Low Bandwidth X (lbx) extension
liblbxutil.so	Utility routines for the Low Bandwidth X (lbx) extension
libmfb.so	Monochrome frame buffer support
libmi.so	Machine-independent portion of the X Server
libmixie.so	With libdixie.so, supports the X Image extensions (XIE)
libos.so	Operating-system dependent portion of the X server
libpanoramiX.so	PanoramiX routines
libprinter.so	Support for the X print server

Table 6–2: Tru64 UNIX Shared Libraries, /usr/shlib/X11 (cont.)

Library /usr/shlib/X11	Description
libxinput.so	X input extension server-side library
libxv.so	Support for the X Video extension
libxvfb.so	Support for the X Virtual Framebuffer extension
libxkb.so	XKEYBOARD extension library

6.4.1 Quickstart

Tru64 UNIX provides Quickstart, which allows shared libraries with unique addresses to start faster than if their addresses were in conflict. Essentially, each shared library must have a unique address placed in the /usr/shlib/so_locations file, which allows applications that link against these shared libraries to start execution faster. This happens because the shared objects do not have to be relocated at run time. The ld utility can read and write an so_locations file when it creates a shared library.

6.4.2 Dynamic Loader

Tru64 UNIX uses a System V Release 4.0 compatible loader to load shared libraries dynamically. This loader provides the following enhanced features:

- The ability to call into dynamically loaded shared libraries
- System V Release 4.0 symbol resolution semantics, including symbol preemption
- The ability to prelink libraries for fast program loading

6.4.3 Versioning

Tru64 UNIX supports full and partial duplication of shared libraries. The loader looks for backward compatible versions of shared libraries using a path constructed by appending the version string as a subdirectory of the normal search path. As a result, any changes to kernel interfaces or to global data definitions that would ordinarily break binary compatibility will not affect your applications, because you can maintain multiple versions of any shared library and link your application against the appropriate version of that shared library.

In Motif Version 1.2, for example, the OSF changed several of the interfaces, thereby breaking binary compatibility with applications built against Motif 1.1.3 libraries. To preserve binary compatibility, Tru64 UNIX

supports both Motif 1.1.3 and Motif 1.2 shared libraries in Tru64 UNIX with our versioning functionality, so that applications that need to can access the Motif 1.1.3 shared libraries. For more information on versioning, see the *Programmer's Guide*.

6.5 Run-Time Libraries

Tru64 UNIX supports the following run-time libraries. With these libraries, you can run previously compiled programs without having to install the corresponding language programs on your system.

- DEC Ada Run-Time Library (`libada`)

  This library supports such DEC Ada run-time functions as tasking, exceptions, timer services, and math functions.

- DEC C++ Run-Time Libraries (`libcxx`, `libcomplex`, and `libtask`)

  These libraries support such DEC C++ run-time functions as I/O handling, complex arithmetic, and multitasking.

- DEC COBOL Run-Time Libraries (`libcob`, `libots2`, and `libisamstub`)

  These libraries support such COBOL run-time functions as I/O handling, decimal arithmetic, COBOL ACCEPT and DISPLAY statements, STRING and UNSTRING operations, and CALL and CANCEL statements.

- DEC Fortran Run-Time Libraries (`libfor`, `libfutil`, and `libUfor`)

  These libraries support such Fortran run-time functions as I/O handling, intrinsic functions, data formatting, data conversion, math functions, and Fortran bindings to common operating system services.

- DEC Pascal Run-Time Library (`libpas`)

  This library supports such Pascal run-time functions as I/O handling, math functions, time and date services, and file services.

6.6 The Java Development Kit

The Java™ Development Kit (JDK) is a component of Tru64 UNIX. You can use this kit to develop and run Java applets and programs.

The JDK for Tru64 UNIX contains a just-in-time compiler (JIT), which provides on-the-fly compilation of your application's Java byte-codes and run-time calls into native Alpha machine code. This results in significantly faster execution of your Java application compared with running it using the Java interpreter. The JIT runs by default when you enter the `java` command.

The JDK implements Java threads on top of native (POSIX) threads. This allows different Java threads in your application to run on different processors, provided that you have a multiprocessor machine. It also means that your Java application will run properly when linked with native methods or native APIs (such as DCE) that are also implemented using POSIX threads.

For more information, see the Java documentation in the following directory on your Tru64 UNIX system where the JDK is installed: `/usr/share/doclib/java/index.html`

6.7 Development Commands

Tru64 UNIX supports the full array of development tools, including `ar`, `as`, `btou`, `cb`, `cc`, `cflow`, `cpp`, `ctags`, `cxref`, `c89`, `dbx`, `dis`, `error`, `file`, `indent`, `ld`, `lex`, `lint`, `loader`, `m4`, `make`, `mig`, `mkstr`, `nm`, `odump`, `pixie`, `ppu`, `prof`, `ranlib`, `size`, `stdump`, `strings`, `strip`, `tsort`, `xstr`, and `yacc`, as well as the source code control systems `rcs` and `sccs`.

Note that many of the development commands are specified by the System V, POSIX, XPG4 and XPG4-UNIX standards to which Tru64 UNIX is fully compliant. Also note that Tru64 UNIX supports both the OSF `make` command and the ULTRIX version of `make`. The ULTRIX `make` command is POSIX 1003.2 compliant and more robust than the OSF `make` command.

6.8 Thread Support

Tru64 UNIX offers DECthreads, Visual Threads, and Thread Independent Services.

6.8.1 DECthreads

The DECthreads library is an implementation of the POSIX 1003.1c-1995 standard multithreading API. DECthreads provides efficient two-level scheduling (POSIX “process contention scope”) that is tightly integrated with the kernel to automatically maintain the maximum level of computational and I/O concurrency at all times. For applications that require real-time scheduling with respect to hardware events or threads in other processes or in the kernel, DECthreads also provides “system contention scope” and scheduling.

6.8.2 Visual Threads

Visual Threads is a tool that lets you debug and analyze multithreaded applications. You can use Visual Threads to automatically diagnose

common problems associated with multithreading; these problems include deadlock, protection of shared data, and thread usage errors.

You can also use this tool to monitor the thread related performance of the application; it will help you to identify bottlenecks or locking granularity problems. It is a unique debugging tool because it can be used to identify problem areas even if an application does not show any specific problem symptoms.

Visual Threads can be used with any Tru64 UNIX application that uses POSIX threads (DECthreads) or is written in Java. It is designed for multithreaded applications of all sizes; it can handle applications with anywhere from two threads to hundreds of threads.

Visual Threads is licensed as part of the Developers' Toolkit.

6.8.3 Thread Independent Services

Tru64 UNIX supports Thread Independent Services (TIS) routines, which are provided to enable application writers to write thread-safe code for nonthreaded libraries and applications. In the presence of threads, these routines provide thread-safe functionality. In the absence of threads, these routines impose the minimum possible overhead on their caller. Note that the TIS routines are used by the C run-time library to provide support for both single and multithreaded applications.

6.9 Memory-Mapped File Support

Tru64 UNIX supports the Berkeley Memory-Mapped File Support (mmap) function, which allows an application to access data files with memory operations rather than file I/O operations.

6.10 Real-Time

Tru64 UNIX supports a real-time user and programming environment; it is shipped as an optional subset. The Tru64 UNIX real-time programming environment conforms to the POSIX 1003.1b-1993 standard for real time, which allows you to develop and run portable real-time applications in a POSIX environment. The POSIX 1003.1b interfaces are collected in the real-time and asynchronous I/O libraries `librt` and `libaio`, respectively.

If you enable kernel preemption, a higher priority process can preempt a lower-priority process regardless of whether it is running in kernel mode or user mode. With this fully preemptive kernel, the Process Preemption Latency (the amount of time it takes to preempt a lower-priority process) is minimized.

In addition to a preemptive kernel, the Tru64 UNIX real-time programming environment supports the following POSIX 1003.1b features:

- Real-time clocks and timers
- Real-time queued signals
- Fixed priority scheduling policies
- Real-time scheduler priority levels
- Counting semaphores
- Shared memory
- Process memory locking
- Asynchronous I/O
- Synchronized I/O
- Message-passing interfaces
- Thread-safe implementation of real-time libraries

For more information on the real-time programming environment, see the *Guide to Realtime Programming*. For information on configuring the real-time kernel, see the *System Administration* guide.

6.11 Networking Program Interfaces

The network programming environment includes the programming interfaces for application, kernel, and driver developers writing network applications and implementing network protocols. Additionally, it includes the kernel-level resources that an application requires to process and transmit data, some of which include libraries, data structures, header files, and transport protocols.

The following programming interfaces are supported by the operating system:

- X/Open Transport Interface (XTI/TLI)
- BSD Sockets
- System V Release 4.0 STREAMS
- Data Link Interface (DLI)
- Data Link Provider Interface (DLPI)
- Extensible SNMP (eSNMP)
- Basic Sockets API for IPv6

For more detailed information on the network programming environment, see the *Network Programmer's Guide*.

6.12 UNIX/NT Interoperability

Tru64 UNIX provides several applications for seamless Windows NT interaction.

Advanced Server for UNIX

Advanced Server for UNIX (ASU) is a layered software application that integrates Tru64 UNIX and Windows environments, providing seamless interoperability between Tru64 UNIX servers, Windows NT servers, and Microsoft Windows clients.

ASU implements Windows NT Version 4.0 Server services and functionality on a system running Tru64 UNIX, making it appear as a Microsoft NT server to other Windows systems. With the ASU software, Tru64 UNIX resources are available to Microsoft users without modification to their software.

You use native Windows commands and utilities to manage the ASU software and to make UNIX based file systems, directories, and printers available to Windows users as shares. Windows users access the shares without modification to their software.

Data Access (ODBC and JDBC)

Tru64 UNIX provides the family of INTERSOLV® DataDirect software products to enable ODBC and JDBC connectivity for your applications. This is optional software for use in developing and deploying applications and is licensed as part of the Tru64 UNIX operating system license.

SequeLink ODBC Edition is a universal ODBC client component. DataDirect SequeLink ODBC provides transparent connectivity to almost any type of client, network, server, or database.

For developers working with Java, JDBC provides Java applications to access data sources and databases across platforms. The SequeLink Java Edition is a universal standards-based implementation of JDBC. It is also flexible in design, providing scalable connectivity from multivendor client, server, and web environments to industry-leading databases. It is optimized and tuned for the Java environment, extending the functionality and performance of existing systems and easily incorporating new technologies.

COM for Tru64 UNIX

COM, the Component Object Model, is middleware developed by Microsoft for the Windows platform. COM implements a binary standard that allows two or more applications to work together regardless of whether they were

written by different vendors, in different languages, at different times, on different platforms running different operating systems. DCOM, the Distributed Component Object Model, extends the COM model and provides applications with a way to interact remotely over a network.

COM for Tru64 UNIX implements Microsoft COM as well as required underlying Windows capabilities to the Compaq Tru64 UNIX platform. The Compaq implementation provides all the basic functions, libraries, and tools that a COM application in a heterogeneous NT client/Tru64 UNIX server environment requires. Programmers who develop on Windows NT only environments will find the same COM Application Programmer Interface (API) and the same behavior in a heterogeneous Windows NT client/Tru64 server environment. COM for Tru64 UNIX provides traditional COM and DCOM capabilities for your application. These capabilities conform to the Microsoft ActiveX Core Technology specification. They include:

- MIDL, the Microsoft Interface Definition Language Compiler that you use to create the component object interface.
- The interfaces and APIs defined by Microsoft as those needed to support COM on non-Windows platforms.
- Support for COM capabilities such as Monikers, OLE Automation, Uniform Data Transfer (UDT), Connectable Objects, and type libraries.
- Multithreaded apartment threading model, formerly known as free threads.
- RPC, Remote Procedure Call, that provides the mechanism for communication across the network.
- Registry, the database of COM components and relevant configuration information, and Registry tools, such as sermon, and regsvr, that allow you to modify Registry contents.
- Security in the form of call security that allows a client or a server to apply an appropriate security level to method calls and Security Support Provider Interface (SSPI) standard that defines security providers that can be accessible to DCOM applications. Microsoft NT uses the Windows NT Distributed Security Provider Interface (also called the NTLM-SSP). COM for Tru64 UNIX supports "pass-through" NTLM SSP calls.
- Internationalization capability, including UNICODE support of wide characters.
- Error handling convention that allow COM objects in different environments to share status information.

6.13 The Windowing Environment

This section briefly discusses the Common Desktop Environment (CDE), and the X11R6.3 and Motif components of the Tru64 UNIX windowing environment.

6.13.1 Common Desktop Environment

CDE is a jointly developed graphical user interface based on industry standards. It is built upon the Open Software Foundation's Motif user interface. CDE provides a consistent look and feel as well as common APIs across multivendor platforms.

CDE presents a visual desktop that you can customize. Using the CDE interface, you can use the mouse or keyboard to navigate and interact with applications. The desktop itself offers a Front Panel, which is a graphical display at the bottom of the screen area that provides access to applications, printers, and frequently used objects, including online help.

In addition to user services, CDE provides everything needed to implement fully integrated applications. Because CDE is standards based, such integration work is transportable to other platforms that comply with this standard. For example, the help files and the means to access them apply across all compliant platforms. For more information, see the *Common Desktop Environment: Programmer's Overview*.

The CDE Front Panel displays the tools that you use to start applications, manage tasks in a desktop session, or change workspaces. Each tool is represented by an icon that indicates its purpose. A workspace is the screen itself, which includes the Front Panel. A tool on the Front Panel is provided to switch between different workspaces.

The tools available on the Front Panel are as follows. For detailed information on the use of each tool, see the *Common Desktop Environment: User's Guide*.

- Clock
- Calendar
- File Manager
- Text Editor
- Mailer
- Lock

- Workspace Switches
- Busy Light
- Printer
- Exit
- SysMan Menu
- Style Manager
- Application Manager
- Help Manager
- Trash Can

CDE Window List is incorporated into the common Desktop Environment; it is invoked by clicking the middle mouse button while the mouse pointer is located in the root window. It finds application windows on any workspace of your desktop and searches for a specific application window given its name, application class, or by the workspace name. For more information, see the online help for CDE Window List.

CDE Setup facilitates the setting up and customization of CDE through a graphical user interface. The CDE Setup utility facilitates the creation of Front Panel controls, which can be assembled into Front Panel types. These types can be customized for different users. Furthermore, the user can set parameters for CDE windows, icons, and bindings, set options for the `dtterm`, `dtmail`, and `dtfile` applications, and determine their Front Panel terminal emulator.

The system administrator has increased capabilities including the configuration of the X server, login greeting, and system services. See the `dtsetup`(8) reference page and the online help for CDE Setup for further information. For more information, see the CDE documentation set.

6.13.2 X Window System

The X11R6.3 windowing software consists of the following components:

- X Client Libraries
- X Server
- Display Manager
- X Protocol Extensions
- Font Server
- X Clients
- X Print Server (Xp)

6.13.2.1 X Client Libraries

Tru64 UNIX Version 5.0 supports the complete set of X11R6.3 X client libraries:

- Athena Widget Set (`libXaw`)

  A high-level library of user-interface components (scroll bars, labels, buttons)

- X Intrinsics Library (`Xt`)

  Middle-level routines that call into `Xlib`

- X library (`Xlib`)

  Low-level routines that interface with the X server

For more information on individual X client libraries, see the X Window documentation in the Tru64 UNIX documentation set.

6.13.2.2 X Server

Through the extensive use of shared libraries, Tru64 UNIX supports a single X11R6.3 X server image for all graphic options. The X server dynamically configures itself during initialization, loading only those server components required by a specific system configuration, and rarely requires any intervention by a system administrator. For a list of the shared libraries that make up the X server, see Table 6–2.

6.13.2.3 Multihead Graphic Support

Multihead graphic support is transparent in Tru64 UNIX, provided the proper option cards are installed and support is built into the kernel. The PanoramiX extension, known as Xinerama in X11R6.4, can be used to take better advantage of multihead systems.

6.13.2.4 X Server Extensions

Tru64 UNIX supports the following X server extensions. Note that, to conserve memory, the X server, by default, defers loading most server extensions until it receives a request from a client for that specific extension.

- The Keyboard Extension for X11R6.3

  The Keyboard Extension for X11R6.3 (XKB) server extension enhances control and customization of the keyboard under the X Window system by providing:

  - Support for the ISO 9996 standard for keyboard layouts
  - Compatibility with the core X keyboard handling; no client modifications needed
  - Standard methods for handling keyboard LEDs and locking modifiers such as CapsLock and NumLock
  - Support for keyboard geometry

  Additionally, the X11R5 (for versions of DIGITAL UNIX earlier than Tru64 UNIX Version 5.0) AccessX server extension for people with physical impairments has been incorporated into the XKB server extension. These accessibility features include StickyKeys, SlowKeys, BounceKeys, MouseKeys, and ToggleKeys, as well as complete control over the autorepeat delay and rate.

- MIT-SHM (MIT Shared memory)

  Enhances performance for local image intensive applications.

- MIT-SUNDRY-nonstandard

  Miscellaneous extension from the X Consortium, which currently controls bug compatibility modes for the X Server.

- Multibuffering

  Supports smooth animations by drawing to multiple buffers.

- SHAPE

  Supports nonrectangular windows used for round, oval, and nonregular shaped windows.

- SMT (Shared Memory Transport)

  Allows for the use of shared memory as an X transport for local clients, giving a significant performance boost. Transport specified by `local:0.0`.

- XIE (X Imaging Extension, Version 3 and 5)

  Provides advanced control over imaging, as well as device-independent image display.

  Tru64 UNIX ships both Version 3 (`/usr/lib/Xie.a` and `/usr/shlib/libXie.so`) and the de facto industry standard, Version 5 (`/usr/lib/libXIE.a` and `/usr/shlib/libXIE.so`).

- X Input

  Allows users to write their own drivers for third-party input devices, and then load them dynamically into the X server by making entries in the X server configuration file (`/usr/var/X11/Xserver.conf`). The new input devices are then recognized the next time the X server is reset.

In traditional, statically linked X Servers, each time a new extension device is added the X Server must be rebuilt. The Tru64 UNIX loadable X server implementation has overcome this limitation by permitting system administrators to add new new input device support as external sharable devices that are loaded by the X server at initialization.

Sample code showing how such a driver should be written is included in the `/usr/examples` directory.

- XKME (X Keyboard Management Extension)

  Provides an internal extension for better support of international X clients. Note that XKME has been made obsolete by the XKB extension, but is provided for backwards compatibility.

- X Screen Saver

  Enables a client to receive notification when the screen has been inactive for a specified amount of time or whenever it cycles. This extension is useful to developers writing screen saver applications.

- XSync

  The `XSync` function, in conjunction with the `XFlush`, `XEventsQueued`, and `XPending` functions, allows synchronization between X clients to take place entirely within the X server, thereby eliminating any errors introduced by the network and enabling different hosts running different operating systems to synchronize X clients. This extension is particularly useful for multimedia applications that require the synchronization of audio, video, and graphics; and for animation applications, which can have their requests synchronized to internal, X server timers.

- XTest

  Allows applications to simulate X events for testing purposes.

- XTrap

  Supports the recording and playback of X events for the purpose of X client testing.

- XV (X Video)

  Allows clients to control video options, such as the live video PIP option for the TX graphic device.

- X Print Extension

  Allows X applications to output directly to a print device.

- X Print Server (Xp)

  An X extension that allows X imaging to nondisplay devices such as printers and fax machines. The core of the X Print service is the X Print Server. Applications that require printing operations can make a

connection to the X Print Server and list the available printers, select a printer, and submit print requests.

- PanoramiX

  Allows a system configured with multiple video monitors (a multiheaded system) to operate the monitors as a single large screen. Windows can span multiple screens and can move from one screen to another. This extension is only supported in homogeneous graphics environments.; the environment must consist of common devices, visuals, depths, resolutions, and so on.

- Application Group Extension

  Allows for grouping of windows such that an application group window can intercept certain requests for windows within its group and handle them appropriately. This is used with the remote execution extension to allow X applications to be embedded in web pages.

- X Security Extension

  Provides for enhanced security for the X Window System. Security can be specified on a per-user basis or a per-resource basis.

- Remote Execution Extension

  Allows X applications to be invoked remotely, specifically from within a web browser.

- Low Bandwidth X (lbx)

  Allows X to run more efficiently over low bandwidth transports such as a modem.

For more information on individual X client libraries, see the X Window documentation in the Tru64 UNIX documentation set and the X(1X) and the Xdec(1X) reference pages.

6.13.2.5 Display Manager

Tru64 UNIX supports the standard Xdm terminal manager software. The Xdm terminal manager starts up the X server locally and allows for network-transparent login prompting, so that users can log in to any system on their network that is supported by xdm as if the remote system's graphic console were in front of them. This software provides for the seamless integration of X terminals into the Tru64 UNIX environment. For more information on using Xdm, see the *System Administration* guide and the xdm(1X) reference page.

Keymap Format

The default keymaps used by Tru64 UNIX use the XKB standard keymap format. These can be used by any X server that supports and runs the XKB extension. The XKB keymap files are text files that easily can be customized and compiled for use with the system.

The `xmodmap` keymap format is also supplied, for backward compatibility. Should you want to run the `xmodmap` keymap format, custom key maps should use this format. The `xmodmap` keymap format is a de facto industry standard, which is written using symbolic key names and can be customized easily.

Both these keymap formats support the ability to specify modifier keys (`Compose`, `Alt`, `Shift`, and so forth).

Compaq suggests using the XKB standard keymap format instead of the `xmodmap` keymap format.

XDM-AUTHORIZATION-1

Whenever an X client application establishes a connection to the X server, it passes an authorization code, called a key, to the X server. If the X server recognizes this key, the connection is allowed. When the user's X session is started, `xdm` (the X Display Manager) writes one or more keys into the `.Xauthority` file in a user's home directory. The X Display Manager (`xdm`) also writes these keys into a file that the X server can read.

To improve security, Tru64 UNIX supports both the `MIT-MAGIC-COOKIE-1` key format as well as the `XDM-AUTHORIZATION-1` encrypted key format, which is the default.

6.13.2.6 Font Server

The operating system supports a standard scalable font server that supplies a network of systems with access to fonts that reside on any Tru64 UNIX system. The font server maintains a repository of fonts and responds to requests from other X servers on the network for fonts that they may not have locally. In addition to providing network-transparent access to fonts, the font server unloads the compute burden of font scaling from local X servers, because it scales fonts appropriately before supplying them to the requesting X server.

Loadable Font Renderers

Before a font can be displayed by an X server, its glyphs must be converted from their on-disk formats into bitmaps. This conversion is done by font

renderer code in the X server or in a font server that may be supplying fonts to the X server.

Tru64 UNIX supports loadable font renderers, so that users who adhere to the X11R6.3 standard can write their own font renderer for their own set of fonts and install them on a Tru64 UNIX system. After the fonts and the font renderer are installed, the necessary entries for them are placed in the X server configuration file (`/usr/var/X11/Xserver.conf`), the font server configuration file (`/usr/var/X11/fs/config`), or in both configuration files. The new font renderer is then recognized the next time the X server or font server (whichever has the font renderer configured) is reset.

6.13.2.7 X Clients

Tru64 UNIX Version 5.0 supports the entire suite of X clients that ships with X11R6.3, including `appres`, `atobm`, `bdftopcf`, `bitmap`, `bmtoa`, `chooser`, `editres`, `fs`, `fsinfo`, `fslsfonts`, `fstobdf`, `iceauth`, `ico`, `imake`, `listres`, `lndir`, `maze`, `mkfontdir`, `oclock`, `optacon`, `puff`, `puzzle`, `restart`, `resize`, `showfont`, `showrgb`, `smproxy`, `twm`, `uil`, `viewres`, `x11perf`, `x11perfcomp`, `xauth`, `xbiff`, `xcalc`, `xcd`, `xclipboard`, `xclock`, `xcmsdb`, `xconsole`, `xcutsel`, `xdm`, `xdpr`, `xdpyinfo`, `xedit`, `xemacs`, `xev`, `xeyes`, `xfd`, `xfindproxy`, `xfwp`, `xfontsel`, `xfwp`, `xhost`, `xkbbells`, `xkbcomp`, `xkbprint`, `xkbdfltmap,`, `xkbvleds`, `xkbwatch`, `xkill`, `xload`, `xlogo`, `xlsatoms`, `xlsclients`, `xlsfonts`, `xmag`, `xman`, `xmbind`, `xmh`, `xmkmf`, `xmodmap`, `xon`, `xpr`, `xprop`, `xrdb`, `xrefresh`, `xset`, `xsetroot`, `xsoundsentry`, `xstdcmap`, `xterm`, `xwd`, `xwininfo`, and `xwud`.

For more information on individual X clients, see the appropriate reference page.

6.13.3 Motif

Tru64 UNIX supports the entire suite of Motif Version 1.0 components, including the widget library (`Xm`), the resource manager (`Mrm`), the widget metalanguage (`wml`), the User Interface Language (UIL), the Motif window manager (`mwm`), the key binding utility (`xmbind`), and the Motif Demonstration programs (`examples`).

In Motif Version 1.2 the OSF added support for ANSI C, Internationalization, Drag and Drop, and TearOff Menus. Unfortunately, much of this support required breaking binary compatibility with Motif Version 1.1.3.

To mitigate this problem, Tru64 UNIX provides the Motif Version 1.1.3 libraries through versioning to allow applications that are linked against

Motif 1.1.3 to continue to run. These libraries are available in an optional subset. For more information on versioning, see the *Programmer's Guide.*

For more information on Motif, see the *OSF/Motif Programmer's Guide* and the appropriate reference pages. For information on Motif support for internationalization, see Chapter 8.

6.13.3.1 Extended Widget Set

Tru64 UNIX supports the Extended Widget Set (`DXm`), which contains the following widgets:

- `DXmColorMix`

  Supports editing and the selection of colors

- `DXmPrintWidget`

  Presents graphical print options

- `DXmCSText`

  Supports the editing of compound strings in a user interface similar to `XmText`

- `DXmHelpWidget`

  Displays help topics

- `DXmSvn`

  Supports structured navigation through lists of data

6.13.3.2 X Clients

In addition to the entire suite of X clients that ships with X11R6.3, Tru64 UNIX supports a variety of X clients including `accessx`, `dxconsole`, `dxdiff`, `dxkbledpabel`, `dxkeyboard`, `dxkeycaps`, `dxpresto`, and `dxterm`.

7

Security

Tru64 UNIX offers a wide range of security configuration options, and can be tailored to the appropriate security level for your installation. The range extends from traditional UNIX security, the default, to the optional enhanced security subsets that, when enabled, satisfy or exceed the requirements of the C2 evaluation class of DoD 5200.28-STD *Trusted Computer System Evaluation Criteria* (TCSEC), also known as the *Orange Book*. A system configured to satisfy these requirements is sometimes referred to as a *trusted system*.

For specific information, see the *Security* guide.

The following sections discuss how Tru64 UNIX addresses the requirements of the C2 evaluation class and additional security features.

7.1 C2 Functions

The *Orange Book* specifies the following C2 requirements, which are all available under the operating system.

- Identification and authentication
- Audit
- Discretionary access controls
- Object reuse
- System architecture
- Integrity
- Security testing
- *Security* guide

7.1.1 Identification and Authentication

The Security Interface Architecture (SIA) allows a single set of identification and authentication (I&A) utilities to work in both base mode (as a *nontrusted system*) and in enhanced mode.

The operating system I&A ships with base mode as the default. Enhanced mode is available through optional security subsets, and the `Security Configuration` leaf of the SysMan menu can enable enhanced mode in one of three ways:

- Shadow passwords only
- Compatibility mode (used for rolling upgrades in TruCluster systems)
- Custom (providing complete configurability)

The SIA allows a single set of identification and authentication utilities to work in either the nontrusted system or the trusted (enhanced security) system. By using the `secsetup` command, you can configure your system to use either nontrusted or enhanced security commands.

The following I&A features are provided in a system running enhanced security:

- Password control
  - Configurable maximum password length is up to 80 characters.
  - Configurable password lifetimes. This includes an optional minimum interval between password changes.
  - A floating value of the minimum password length, based directly on the *Department of Defense Password Management Guideline* (*Green Book*) guidelines and the password lifetime.
  - Per user password generation flags, which include the ability to require a user to have a generated password.
  - Recording of who (besides the user) last changed the user's password.
- Login control
  - Optional recording of the last terminal and time of the last successful login, and of the last unsuccessful login attempt.
  - Automatic account lockout after a specified number of consecutive bad access attempts (breakin detection and evasion).
  - A per-terminal setting for the delay between consecutive login attempts and the maximum amount of time each attempt is allowed before being declared a failed attempt.
  - A per-terminal setting for maximum consecutive failed login attempts before locking any new accesses from that terminal.
- Differentiation between "retired" and "locked" accounts.
- Configurable multilevel system default values for the various I&A fields (templates).

- A CDE-based GUI (`dxaccounts`) to perform user account management and other I&A administration tasks.

  The `edauth`, `convauth`, and `convuser` utilities to ease the migration of accounts to the enhanced security level.

7.1.2 Audit Features

The following audit features are provided:

- An audit configuration GUI, `auditconfig`, to ease system auditing
- Command line interfaces
- The ability to send audit logs to a remote host
- Fine-grained preselection of system events, application events, and site-definable events
- Event profiles allowing simplified configuration based on system type
- Fine-grained post-analysis of system events, application events, and site-definable events
- Optional automated audited log cleanup
- The `dxaudit` GUI

When used with enhanced security, auditing additionally includes support for a per-user audit characteristics profile with enhanced identification and authorization. The audit system is set up through the `Audit Configuration` leaf of the SysMan menu or by using the `auditconfig` utility from the command line. Maintenance for the audit subsystem is done from the command line or with the `dxaudit` GUI.

7.1.3 Discretionary Access Controls

Discretionary access controls (DACs) provide the capability for users to define how the resources they create can be shared. The traditional UNIX permission bits provide this capability.

Tru64 UNIX also provides optional access control lists (ACLs) to provide object protection at the individual user level. ACLs are supported under the UFS, NFS, AdvFS, and CFS file systems. To simplify ACL management, an ACL GUI, named `dxsetacl` is available in addition to the command line interface.

7.1.4 Object Reuse

Object reuse ensures that the physical storage (memory or disk space) assigned to shared objects or physical storage that is released prior to

reassignment to another user, is cleared or “scrubbed.” Examples of object reuse are disk space that is released after a file is truncated or physical memory that is released prior to reassignment to another user to read.

7.1.5 Protected Environment for Trusted Components

The operating system uses hardware memory management to maintain a kernel address space for itself and to maintain separate address spaces for each instance of an executing application process. Processes may try to write to the same address space. DACs control the sharing of this address space among processes; the default is to disallow sharing. The administrator can disable the sharing of sections as read-only address space, for example, shared libraries. Thus, the security-relevant components of the system (the trusted computing base, or TCB) are protected while they execute.

The operating system protects the on-disk security components using discretionary access control. Attempted violations of the DAC protections can be audited so that remedial action can be taken by the system security officer.

In addition, the security components are structured into well defined, largely independent modules.

The operating system is designed, developed, and maintained under a configuration management system that controls changes to the specifications, documentation, source code, object code, hardware, firmware, and test suites. Tools, which are also maintained under configuration control, are provided to control and automate the generation of new versions of the security components from source code and to verify that the correct versions of the source have been incorporated into the new version.

The master copies of all material used to generate the security components are protected from unauthorized modification or destruction.

7.1.6 Integrity Features

The operating system provides the capability to validate the correct operation of hardware, firmware, and software components of the security components. The firmware includes power-on diagnostics and more extensive diagnostics that optionally can be enabled. The firmware itself resides in EEPROM and can be physically write-protected. It also can be compared against, or reloaded from, an offline master copy. Additional hardware diagnostics can be used also.

The firmware can require authorization to load any operating software other than the default or to execute privileged console monitor commands that examine or modify memory.

Once the operating system is loaded, administrators can run system diagnostics to validate the correct operation of the hardware and software. In addition, test suites are available to ensure the correct operation of the operating system software.

The following tools can be run automatically to detect inconsistencies in the security software and databases:

- `fverify`

  This command reads subset inventory records from standard input and verifies that the attributes for the files on the system match the attributes listed in the corresponding records. Missing files and inconsistencies in file size, checksum, user ID, group ID, permissions, and file type are reported.

- `authck`

  This program checks both the overall structure and internal field consistency of all components of the authentication database and reports all problems that it finds.

7.1.7 Security Administration

The operating system provides system administrators with tools to improve the ease of use of administering system security.

7.1.7.1 Configuring System Security

System administrators can select the security level associated with their system. The default base security level consists of object reuse and DAC; through the `Security Configuration` leaf of the SysMan menu or by running `secconfig` from the command line, system administrators can select enhanced security features and ACLs. The audit subsystem is configurable either through the `Audit Configuration` leaf of the SysMan menu or by using the `auditconfig` utility from the command line.

7.1.7.2 Windows-Based Administration Utilities

Tru64 UNIX provides four GUIs to deal with day-to-day security administration on the local machine:

- The `dxaccounts` utility (Account Manager under the CDE-based system administration utilities) operates in both base and enhanced security modes and is used to create and maintain selected system

defaults. It provides the ability to create template accounts, and to modify selected system defaults. Under enhanced security, it allows a specification of a per-user set of events to audit (audit mask).

- The `dxaudit` utility controls the administration of the audit system (for instance, defining which events to audit) and the generation of audit reports. Auditing is available with both base and enhanced security.
- The `dxsetacl` utility (ACL Manager under the CDE based system administration utilities) is used to create, modify, and delete ACLs on files and directories.
- The `dxdevices` utility configures secure devices when running in enhanced security mode.

The `audit_setup`, `secsetup`, `XSysAdmin`, and `XIsso` utilities have been retired. They are available in the Obsolete subset, but their use is not recommended.

For more information, see the `dxaccounts`(8X), `dxsetacl`(8X), `dxaudit`(8X), and `dxdevices`(8X) reference pages.

7.2 Other Security Features

Tru64 UNIX supports some features not available in other OSF-based UNIX operating systems.

7.2.1 Features of the Security Integration Architecture Layer

All security mechanisms that run on the Tru64 UNIX operating system run under the Security Integration Architecture (SIA) layer. The SIA allows a suitably privileged administrator to layer various local and distributed security authentication mechanisms onto Tru64 UNIX with no modification to the security-sensitive Tru64 UNIX commands, such as `login`, `su`, and `passwd`. The SIA isolates the security-sensitive commands from the specific security mechanisms, thus eliminating the need to modify them for each new security mechanism.

Because of the presence of the SIA, administrators can use the `secconfig` command to toggle back and forth between the secure and nonsecure commands and utilities.

7.2.1.1 Network Information Service Compatibility

Tru64 UNIX provides support for accessing Network Information Service (NIS) distributed databases while running enhanced security. For example, administrators can use the `ypcat passwd` command to gather information

about users on the network. However, the user's encrypted password in the NIS distributed password database is not the same as the encrypted password on the secure system, which cannot be viewed by unprivileged users.

In addition, on a Tru64 UNIX system running enhanced security, NIS can be used to distribute the enhanced security protected password database.

7.2.1.2 DECnet Interoperability

The SIA interface provides support for the DECnet/OSI networking software.

7.2.1.3 Distributed Computing Environment Interoperability

Through the SIA, Tru64 UNIX, when configured for enhanced security, allows you to enter both your system password and your Distributed Computing Environment (DCE) password at login time. You do not have to log in to the Tru64 UNIX secure system and then log in again to DCE.

7.2.2 Configuration and Setup Scripts

Tru64 UNIX supports the `secconfig` configuration and setup script, which allows you to select the security level you want to run, permits you to toggle back and forth between secure and nonsecure commands and utilities, and configures security at boot time, depending upon the value of the `SECURITY` variable in the `/etc/rc.config` file.

7.2.3 Graphical User Interfaces

Tru64 UNIX provides the `dxaccounts`, `dxaudit`, and `dxdevices` utilities that permit the creation and modification of user accounts, modification of system defaults, and all the audit interfaces and devices. The `dxsetacl` utility is provided for users to manipulate ACLs on system objects.

7.2.4 Division of Privileges

The Division of Privileges (DOP) feature of Tru64 UNIX is designed to provide role-based privilege management. Using the DOP configuration utility, `dopconfig`, a system administrator can assign users or groups the privilege to execute a set of actions that would usually be reserved for root.

8

Internationalization

8.1 Overview

The term “internationalization” is formally defined by The Open Group as a “provision within a computer program of the capability of making itself adaptable to the requirements of different native languages, local customs, and coded character sets,” which means, essentially, that internationalized programs can run in any supported **locale** without having to be modified. A locale is a software environment that correctly handles the cultural conventions of a particular geographic area, such as China or France, and a language as it is used in that area. So by selecting a Chinese locale, for example, all commands, system messages, and keystrokes can be in Chinese characters and displayed in a way appropriate for Chinese.

Tru64 UNIX is an internationalized operating system that not only allows users to interact with existing applications in their native language, but also supports a full set of application interfaces, referred to as the Worldwide Portability Interfaces (WPI), to enable software developers to write new internationalized applications. The original code for these interfaces came from the Open Software Foundation (OSF) and has been enhanced.

The internationalization support in the operating system conforms to The Open Group’s CAE specifications for system interfaces and headers (XSH Issue 5), curses (XCURSES Issue 4.2), and commands and utilities (XCU Issue 5). These specifications align with current versions of the POSIX and ISO C standards. This conformance ensures that commands, utilities, and libraries have been internationalized, and their corresponding message catalogs have been included in the base system. In addition, the operating system supports the X Input Method (XIM) and X Output Method(XOM) to facilitate input of local language characters, text drawing, measurement, and interclient communication. These functions are implemented according to the X11R6.3 specification and include some problem corrections specified by X11R6.4.

Note that the operating system also supports a 32 bit `wchar_t` datatype which in turn enables support for a wide array of codesets, including the one defined by the full ISO 10646 standard.

For more comprehensive information about internationalization features discussed in this section, see *Writing Software for the International Market*.

8.2 Supported Languages

Table 8–1 lists the languages supported by the operating system and their corresponding locales. Most locales are included in Worldwide Language Support (WLS) subsets that are optionally installed. Some, as indicated in the table, are part of the mandatory base operating system.

Locale names that include `@ucs4` and `UTF-8` support character encoding as defined by the ISO 10646 standard. The `wchar_t` data type is 32 bits in length, with zero padding of leading bits for those character values that do not require all 32 bits. The first 256 values of the Universal Character Set define the same characters as defined for the ISO 8859–1 (Latin-1) character set. Therefore, the Tru64 UNIX implementation of the `wchar_t` type is identical to UCS-4 process code for all `ISO8859--1` locales. `ISO8859--1` locales differ from `UTF-8` locales with the same base name in terms of data file encoding and the fact that only the `UTF-8` locales support the euro monetary character.

The English locale name that includes `cp850` supports character encoding in PC code-page format.

For the most up-to-date list of supported languages and locales, refer to the `l10n_intro`(5) reference page. This book may not be updated for minor functional releases of the operating system and locales are sometimes added for such releases.

Table 8–1: Languages and Locales

Language	Locale Name
Catalan	`ca_ES.ISO8859-1` [a] `ca_ES.UTF-8`

Simplified Chinese (PRC)	`zh_CN.dechanzi`
	`zh_CN.dechanzi@ucs4`
	`zh_CN.dechanzi@pinyin`
	`zh_CN.dechanzi@pinyin@ucs4`
	`zh_CN.dechanzi@radical`
	`zh_CN.dechanzi@radical@ucs4`
	`zh_CN.dechanzi@stroke`
	`zh_CN.dechanzi@stroke@ucs4`

Table 8–1: Languages and Locales (cont.)

Language	Locale Name
Traditional Chinese (Hong Kong)	zh_HK.big5 zh_HK.dechanyu zh_HK.dechanyu@ucs4 zh_HK.dechanzi zh_HK.dechanzi@ucs4 zh_HK.eucTW zh_HK.eucTW@ucs4
Traditional Chinese (Taiwan)	zh_TW.big5 zh_TW.big5@chuyin zh_TW.big5@radical zh_TW.big5@stroke zh_TW.dechanyu zh_TW.dechanyu@ucs4 zh_TW.dechanyu@chuyin zh_TW.dechanyu@chuyin@ucs4 zh_TW.dechanyu@radical zh_TW.dechanyu@radical@ucs4 zh_TW.dechanyu@stroke zh_TW.dechanyu@stroke@ucs4 zh_TW.eucTW zh_TW.eucTW@ucs4 zh_TW.eucTW@chuyin zh_TW.eucTW@chuyin@ucs4 zh_TW.eucTW@radical zh_TW.eucTW@radical@ucs4 zh_TW.eucTW@stroke zh_TW.eucTW@stroke@ucs4
Czech	cs_CZ.ISO8859-2 cs_CZ.ISO8859-2@ucs4
Danish	da_DK.ISO8859-1 [a] da_DK.UTF-8
Dutch	nl_NL.ISO8859-1 [a] nl_NL.UTF-8
Belgian Dutch	nl_BE.ISO8859-1 [a] nl_BE.UTF-8
US English/ASCII	C (POSIX) [a]
US English	en_US.ISO8859-1 [a] en_US.cp850. en_US.UTF-8, en_US.UTF-8@euro [b]
GB English	en_GB.ISO8859-1 [a] en_GB.UTF-8
European	en_EU.UTF-8@euro [c]

Table 8–1: Languages and Locales (cont.)

Language	Locale Name
Finnish	fi_FI.ISO8859-1 [a] fi_FI.UTF-8
French	fr_FR.ISO8859-1 [a] fr_FR.UTF-8
Belgian French	fr_BE.ISO8859-1 [a] fr_BE.UTF-8
Canadian French	fr_CA.ISO8859-1 [a] fr_CA.UTF-8
Swiss French	fr_CH.ISO8859-1 [a] fr_CH.UTF-8
German	de_DE.ISO8859-1 [a] de_DE.UTF-8
Swiss German	de_CH.ISO8859-1 [a] de_CH.UTF-8
Greek	el_GR.ISO8859-7, el_GR.ISO8859-7@ucs4 el_GR.UTF-8
Hebrew	he_IL.ISO8859-8 he_IL.ISO8859-8@ucs4 iw_IL.ISO8859-8 iw_IL.ISO8859-8@ucs4
Hungarian	hu_HU.ISO8859-2 hu_HU.ISO8859-2@ucs4
Icelandic	is_IS.ISO8859-1 [a]
Italian	it_IT.ISO8859-1 [a] it_IT.UTF-8
Japanese	ja_JP.eucJP ja_JP.SJIS ja_JP.SJIS@ucs4 ja_JP.deckanji ja_JP.deckanji@ucs4 ja_JP.sdeckanji
Korean	ko_KR.deckorean ko_KR.deckorean@ucs4 ko_KR.eucKR ko_KR.KSC5601
Lithuanian	lt_LT.ISO8859-4 lt_LT.ISO8859-4@ucs4
Norwegian	no_NO.ISO8859-1 [a] no_NO.UTF-8

Table 8–1: Languages and Locales (cont.)

Language	Locale Name
Polish	pl_PL.ISO8859-2 pl_PL.ISO8859-2@ucs4
Portuguese	pt_PT.ISO8859-1 [a] pt_PT.UTF-8
Russian	ru_RU.ISO8859-5 ru_RU.ISO8859-5@ucs4
Slovak	sk_SK.ISO8859-2 sk_SK.ISO8859-2@ucs4
Slovene	sl_SI.ISO8859-2 sl_SI.ISO8859-2@ucs4
Spanish	es_ES.ISO8859-1 [a] es_ES.UTF-8
Swedish	sv_SE.ISO8859-1 [a] sv_SE.UTF-8
Thai	th_TH.TACTIS
Turkish	tr_TR.ISO8859-9 tr_TR.ISO8859-9@ucs4

[a]This locale is included in the base installation rather than in an optional worldwide language support subset.
[b]The en_US.UTF-8@euro locale defines the local currency sign to be the euro character and the international currency sign to be EUR.
[c]The en_EU.UTF-8 locale defines the local currency sign to be the euro character, the international currency sign to be EUR, the decimal point to be a comma (,), and the thousands separator to be a period (.). When assigned specifically to the LC_MONETARY locale category or environment variable, this locale is useful in many European countries, not just those in which English is spoken.

Note that you can switch languages or character sets as necessary and can even operate multiple processes in different languages or codesets in the same system at the same time.

For more information on a particular coded character set, such as ISO8859-9, see the reference page with the same name. For more information about UCS-4 and UTF-8 encoding, see Unicode(5). For more information about PC code pages, see code_page(5).

8.3 Enhanced Terminal Subsystem for Asian Languages

The base tty terminal driver subsystem is extended to include additional BSD line disciplines and STREAMS terminal driver modules for processing data in Chinese, Japanese, Korean, and Thai. For example, the enhanced terminal subsystem supports the following capabilities for these languages:

- Japanese Kana-Kanji conversion input method
- Character-based line processing in cooked mode

- Input line history and editing (BSD line discipline only)
- Software on-demand-loading for user-defined characters
- Conversion between terminal code and application code

For more information about the Asian and Thai terminal subsystems, see the atty(7), tty(7), and stty(1) reference pages.

8.4 Codeset Conversion

The operating system includes the iconv utility and the iconv_open(), iconv(), and iconv_close() functions, which convert text from one codeset to another, thereby assisting programmers in the writing of international applications. For use with these interfaces, the operating system includes a large set of codeset converters. In addition to conversion between different codesets for the same language, these converters support conversion between different Unicode formats, such as UCS-2, UCS-4, and UTF-8. There are also codeset converters that handle the most commonly used PC code-page formats.

Codeset conversion is also used by the printing subsystem and utilities, such as man, to allow processing of files in different languages and encoding formats. Additionally, codeset conversion is implemented in mail utilities for mail interchange with systems using different codesets and in the X Window Toolkit for text input, drawing, and interclient communication. For more information on codeset conversion, see the iconv_intro(5) reference page. See the Unicode(5) and code_page(5) reference pages for a discussion of converters for Unicode and PC code pages, respectively.

8.5 Unicode Support

The operating system provides both codeset converters and locales that support the Unicode and ISO 10646 standards. The codeset converter modules convert between other supported codesets and UCS-2, UCS-4, and UTF-8 formats. In addition to the country- and language-specific locales listed in Section 8.2, there is also the universal.UTF-8 locale that programmers can use to process characters in all languages by using UCS-4 encoding format.

The operating system provides a function called fold_string_w(), which maps one Unicode string to another and performs the specified Unicode transformations. For more information on the fold_string_w() function, see fold_string_w(3). For more information on Unicode support, see Unicode(5).

8.6 Support for the euro Character

The operating system supports the new euro currency now being used by member countries of the European Economic and Monetary Union (EMU).

Locales that use the UTF-8 codeset support the euro character. Those whose names include the `@euro` suffix also define the local currency sign to be the euro character and the international currency sign to be `EUR`. See Section 8.2 for more information about locales.

The ISO Latin-9 codeset (`ISO8859-15`), which also includes the euro character, forms the basis of euro font support. The operating system includes both screen and PostScript outline fonts for this codeset. See `ISO8859-15`(5) for a list of these fonts. The operating system does not provide native Unicode fonts that support the euro character. However, the X font library has been extended to combine a number of fonts together to form logical Unicode fonts for applications to use. The names of these logical fonts include the string `ISO10646-1`.

Printer support for the euro character is enabled by a generic PostScript print filter, `wwpsof`, which supports printing of file data in UTF-8 format. See `wwpsof`(8) for information on setting up printers with this print filter.

Keyboard entry of the euro character is supported by language- and keyboard-specific key sequences that are defined in keymaps (XKB format). The euro character also can be entered by using a Compose key sequence on those keyboards that support a Compose key. The `euro`(5) reference page lists these key sequences.

Finally, the operating system provides codeset converters to convert file data between the various encoding formats that support the euro character. Specifically, codeset converters can convert file data between:

- Unicode encoding formats and PC code-page formats that support the euro
- Unicode encoding formats and `ISO8859-15` encoding

See `euro`(5) for a more detailed discussion of the information in this section.

8.7 Internationalized Curses Library

The operating system supplies an internationalized Curses library in conformance with X/Open Curses, Issue 4 Version 2. This library provides functions for processing characters that span one or multiple bytes. These characters may be in either wide-character (`wchar_t`) or complex-character

(cchar_t) formats. The complex-character format provides for a single logical character made up of multiple wide characters. Some of the components of the complex character may be nonspacing characters.

For information on the syntax and effect of all Curses interfaces, see curses(3). For a description of the enhancements provided by the internationalized Curses routines, and their relationship to previous Curses routines, see *Writing Software for the International Market*.

8.8 Internationalized Printing

The operating system supports the printing of plain text and PostScript files for a variety of languages and provides outline fonts for high quality printing on PostScript printers. In addition to print filters for a variety of local-language printers, there are also generic internationalized print filters that can be used with both Compaq and third-party printers. One of these filters, wwpsof, supports printing of local-language files on PostScript printers that do not include the required fonts. For more information on internationalized printing features, see the i18n_printing(5), pcfof(8), and wwpsof(8) reference pages.

8.9 Locale Creation

The localedef utility allows programmers to create their own locales, compile their source, and generate a unique name for their new locale.

For more information on creating locales, see *Writing Software for the International Market*.

8.10 I18N Configuration Tool

The I18N Configuration tool, available through the Application Manager, is one of the CDE System Administration Configuration applications. I18N Configuration provides a graphical interface for the system administrator to configure I18N-specific settings. It also provides a convenient way to see which countries, locales, fonts, and keymaps are supported on the host system. System administrators can also use this tool to remove unused fonts and country support from the system.

8.11 Enhanced Sorting for Asian Languages

The operating system supports the asort utility, an extension of the sort command, which allows characters of ideogrammatic languages, like Chinese and Japanese, to be sorted according to multiple collation sequences. For more information on the asort utility, see asort(1).

8.12 Multilingual EMACS Editor

The operating system supports the Multilingual EMACS editor (MULE) for Asian languages. See `mule`(1) for more information.

8.13 Mail and 8-Bit Character Support

By default, the operating system provides support for 8–bit character encoding in `mailx`, `dtmail`, `MH`, and `comsat`.

For more information on these mail utilities, see `mailx`(1), `dtmail`(1), `mh`(1), and `comsat`(8).

8.14 Enhanced file command

The `file` (see `file`(1)) is enhanced to recognize UCS-2 and UCS-4 encoding in any locale setting. For other encoding formats, the command recognizes file data encoding if it is valid for the current locale setting. This command also has a `jfile` alias that, in any locale, can recognize DEC Kanji, Japanese EUC, Shift JIS, and 7–bit JIS encoding.

8.15 Support for User-Defined Characters

The operating system provides support for creating user-defined characters (UDCs) in Chinese, Japanese, and Korean, so that users can create and define character fonts and their attributes, including bitmap fonts, with the `cedit` and `cgen` utilities.

Font rendering facilities are available so that X clients can use UDC databases through the X Server or font server to obtain bitmap fonts for user-defined characters.

For more information on user-defined characters, see *Writing Software for the International Market*, `cedit`(1) and `cgen`(1).

8.16 Internationalization for Graphical Applications

Motif Version 1.2.3 takes advantage of many of the internationalization features of X11R6 and the C library to support locales. Motif Version 1.2.3 also supports the use of alternate input methods, which allows input of non-ISO Latin-1 keystrokes, and delivers an extensively rewritten `XmText` widget which supports multibyte and wide character format and on-the-spot input style.

Motif supports multibyte and wide-character encoding through the use of the internationalized X library functions, and C Library functions. In addition, the compound string routines include the X11R6 `XFontSet` component to allow for the creation of localized strings.

The User Interface Language (UIL) supports the creation of localized UID files through the `-s` compile-time switch on the UIL compiler, which causes the compiler to construct localized strings.

Alternate input methods can be specified by a resource on the `VendorShell` widget. Widgets that are parented by a `Shell` class widget can take advantage of this resource and register themselves as using a specific method for input.

The following sections discuss internationalization features of Motif widgets and internationalized client applications.

8.16.1 Internationalized Motif Widgets

The following lists contain the widgets in the Motif Toolkit and in the Extensions to the Motif Toolkit that support local language characters I/O capabilities and local language message displays.

Note that the Motif UIL compiler is extended to support local language characters in UIL files.

- Motif Toolkit
  - Command
  - FileSelectionBox
  - Label
  - MessageBox
  - SelectionBox
  - Text
  - TextField
- Extensions to Motif Toolkit
  - ColorMix
  - CSText
  - Help
  - Print
  - Structured Visual Navigation (SVN)

8.16.2 Internationalized CDE Clients

CDE is the default desktop for Tru64 UNIX. The following CDE clients are internationalized:

- Application Manager
- Calculator
- Calendar
- Create Action
- File Manager
- Front Panel
- Help Viewer
- Icon Editor
- Login Screen
- Message
- Mailer
- Print Manager
- Style Manager
- Terminal Emulator
- Trash Can

By default, client applications run in the language set by the user at the start of a CDE session. However, users can also change locale in a terminal emulation window and invoke an application in a language different from the session default.

8.16.3 Additional Internationalized Motif Clients

The operating system includes the following internationalized clients in addition to those common to all CDE implementations:

- Differences
- Keycap
- DECterm
- X Display Manager

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

How to Order Tru64 UNIX Documentation

You can order documentation for the Tru64 UNIX operating system and related products at the following Web site:

http://www.businesslink.digital.com/

If you need help deciding which documentation best meets your needs, see the Tru64 UNIX *Documentation Overview* or call **800-344-4825** in the United States and Canada. In Puerto Rico, call **787-781-0505**. In other countries, contact your local Compaq subsidiary.

If you have access to Compaq's intranet, you can place an order at the following Web site:

http://asmorder.nqo.dec.com/

The following table provides the order numbers for the Tru64 UNIX operating system documentation kits. For additional information about ordering this and related documentation, see the *Documentation Overview* or contact Compaq.

Name	Order Number
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
Startup Documentation Kit	QA-6ADAC-GZ
General User Documentation Kit	QA-6ADAD-GZ
System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAG-GZ
Reference Pages Documentation Kit	QA-6ADAF-GZ

Reader's Comments

Tru64 UNIX
Technical Overview
AA-RH8XA-TE

Compaq welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

- This postage-paid form
- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: (603) 884-0120, Attn: UBPG Publications, ZKO3-3/Y32

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Usability (ability to access information quickly)	☐	☐	☐	☐

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____
_____
_____
_____

What version of the software described by this manual are you using? _____

Name, title, department _____
Mailing address _____
Electronic mail _____
Telephone _____
Date _____

--------- Do Not Cut or Tear - Fold Here and Tape ----------

COMPAQ

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ COMPUTER CORPORATION
UBPG PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK RD
NASHUA NH 03062-9987

--------- Do Not Cut or Tear - Fold Here ----------

Cut on Dotted Line