

# **Release Notes for HP-UX 10.20**

## **HP 9000 Computers**



**Printed in USA June 1996**

**Fourth Edition  
E0696**

---

## Legal Notices

The information in this document is subject to change without notice.

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Warranty.** A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

**Restricted Rights Legend.** Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY  
3000 Hanover Street  
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

**Copyright Notices.** ©copyright 1983-96 Hewlett-Packard Company, all rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

©copyright 1979, 1980, 1983, 1985-93 Regents of the University of California

This software is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

©copyright 1980, 1984, 1986 Novell, Inc.

©copyright 1986-1992 Sun Microsystems, Inc.

©copyright 1985-86, 1988 Massachusetts Institute of Technology.

©copyright 1989-93 The Open Software Foundation, Inc.

©copyright 1986 Digital Equipment Corporation.

©copyright 1990 Motorola, Inc.

©copyright 1990, 1991, 1992 Cornell University

©copyright 1989-1991 The University of Maryland

©copyright 1988 Carnegie Mellon University

**Trademark Notices** UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

MS-DOS and Microsoft are U.S. registered trademarks of Microsoft Corporation.

OSF/Motif is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries.

---

## Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: January 1995 (HP-UX Release 10.0)

Second Edition: June 1995 (HP-UX Release 10.0 version B.10.01)

Third Edition: November 1995 (HP-UX Release 10.10)

Fourth Edition: June 1996 (HP-UX Release 10.20)

# Contents

---

<b>1. Overview of Release Notes for HP-UX 10.20</b>	
The Release Notes and Release Notes Files . . . . .	1-2
The Purpose of Release Notes . . . . .	1-2
Other Sources of Information . . . . .	1-3
README Documents . . . . .	1-3
Online Documents . . . . .	1-3
HP-UX Upgrade Tools Documents . . . . .	1-3
The HP-UX Documentation Set . . . . .	1-4
<b>2. Major Changes for HP-UX 10.20</b>	
64-bit Data Register . . . . .	2-2
Summary of Change . . . . .	2-2
Impact . . . . .	2-3
Performance . . . . .	2-3
Compatibility . . . . .	2-4
CDE (HP Common Desktop Runtime Environment) . . . . .	2-4
Pluggable Authentication Modules (PAM) . . . . .	2-4
Impact . . . . .	2-5
Configuration . . . . .	2-5
Known Problems . . . . .	2-5
Vacation Mail . . . . .	2-5
Error Messages . . . . .	2-6
HP CDE and Single Logical Screen (SLS) Server . . . . .	2-6
Cold Install . . . . .	2-8
DHCP (Dynamic Host Configuration Protocol) . . . . .	2-9
Impact . . . . .	2-9
Performance . . . . .	2-10
Alternatives . . . . .	2-10
Configuration . . . . .	2-10
HP Distributed Print Service (HPDPS) . . . . .	2-10

Impact . . . . .	2-12
Size Requirement . . . . .	2-12
Additional Information . . . . .	2-12
HP-UX Bootstrap . . . . .	2-13
JFS Version 3 . . . . .	2-13
Impact . . . . .	2-14
Compatibility . . . . .	2-14
Alternatives . . . . .	2-14
JFS as the Root File System . . . . .	2-15
/stand Directory . . . . .	2-15
Rebooting . . . . .	2-16
Mirroring . . . . .	2-16
Boot Messages . . . . .	2-16
System Recovery . . . . .	2-16
Local Root File Systems . . . . .	2-17
User-Space Startup Software . . . . .	2-17
Kernel Support . . . . .	2-17
Compatibility . . . . .	2-17
Large Files . . . . .	2-18
Interfaces Supporting Large Files . . . . .	2-18
Data Types . . . . .	2-18
Data Structures . . . . .	2-19
Subroutine Interfaces . . . . .	2-20
New Interfaces . . . . .	2-20
Commands . . . . .	2-21
Behavior of Existing Applications . . . . .	2-21
Programming for Large Files . . . . .	2-22
Using Large Files . . . . .	2-23
EOVERFLOW Error . . . . .	2-24
Commands Support . . . . .	2-24
Shells Support . . . . .	2-24
Backing Up Large Files . . . . .	2-24
Performance . . . . .	2-25
Compatibility . . . . .	2-25
Large User IDs . . . . .	2-25
Impact . . . . .	2-26
Limitations . . . . .	2-29
Performance . . . . .	2-30

Alternatives/Compatibility . . . . .	2-30
Obsolescence . . . . .	2-31
Multi-Processor Control . . . . .	2-31
QIC Media . . . . .	2-31
Reverse ARP (Address Resolution Protocol) . . . . .	2-32
Impact . . . . .	2-32
Compatibility . . . . .	2-32
SD-UX . . . . .	2-32
Enhancements and Defect Fixes . . . . .	2-34
Impact . . . . .	2-34
Performance . . . . .	2-35
Compatibility . . . . .	2-35
System Compatibility . . . . .	2-35
VM Kernel Address Space . . . . .	2-36
VME Services . . . . .	2-38
Performance . . . . .	2-38
Supportability . . . . .	2-38
Installability . . . . .	2-39
Binary Compatibility . . . . .	2-39
Standards . . . . .	2-39

### 3. Major Changes for HP-UX 10.10

System Compatibility . . . . .	3-3
Upgrading from HP-UX 9.x to 10.x . . . . .	3-3
Compatibility with Systems Before HP-UX 10.10 . . . . .	3-4
System Configuration . . . . .	3-5
New System Configurations (S700/800): . . . . .	3-5
Pre-enabled Hardware . . . . .	3-5
Supported NFS Diskless Configurations . . . . .	3-5
Product Stability Notice . . . . .	3-6
Large File Systems . . . . .	3-7
Making Small File Systems Large . . . . .	3-7
Compatibility with pre-10.10 Filesystems . . . . .	3-7
Limitations . . . . .	3-7
Performance . . . . .	3-8
dcopy Command . . . . .	3-8
fsirand . . . . .	3-8
Large Physical Memory (3.75 Gb) . . . . .	3-9

Impact . . . . .	3-11
Performance . . . . .	3-11
Large Process Data Space . . . . .	3-13
Impact . . . . .	3-14
Performance . . . . .	3-14
Alternatives/Compatibility . . . . .	3-14
Native Language Support (NLS) . . . . .	3-15
Impact . . . . .	3-16
Performance . . . . .	3-16
Compatibility . . . . .	3-16
Alternatives . . . . .	3-16
Localization . . . . .	3-16
Future MSE Changes . . . . .	3-17
Compatibility . . . . .	3-17
Locale cs_CZ.iso88592 Change . . . . .	3-18
3D Graphics . . . . .	3-18
Upgrading from 9.x . . . . .	3-18
PCx-Processor Support Pre-Enable . . . . .	3-19
Configuration . . . . .	3-19
Distributed Interrupts and I/O Forwarding . . . . .	3-20
Impact/Performance . . . . .	3-21
Limitations . . . . .	3-21
Logical Volume Manager (LVM) . . . . .	3-21
Shared Logical Volume Manager (SLVM) . . . . .	3-22
Performance . . . . .	3-24
Compatibility . . . . .	3-24
Upgrade from 9.04 . . . . .	3-24
UNIX95 . . . . .	3-24
Modified Interfaces . . . . .	3-25
fstat(), lstat(), and stat() . . . . .	3-25
ftruncate() and truncate() . . . . .	3-25
getitimer() and setitimer() . . . . .	3-25
getrlimit() and setrlimit() . . . . .	3-25
gettimeofday() . . . . .	3-26
mmap() . . . . .	3-26
mprotect() . . . . .	3-26
msync() . . . . .	3-26
munmap() . . . . .	3-27



readlink()	3-27
readv() and writev()	3-27
select()	3-27
setpgrp()	3-28
sigignore(), sigrelse(), sighold, sigpause(), and sigset()	3-28
sigstack()	3-28
waitid(), waitpid(), and wait3()	3-29
New Interfaces	3-29
getcontext() and setcontext()	3-29
getdtablesize()	3-29
gethostid()	3-30
getpagesize()	3-30
getrusage()	3-30
makecontext() and swapcontext()	3-30
setreuid()	3-31
sigaltstack()	3-31
ualarm()	3-31
usleep()	3-31
utimes()	3-32
Signal Changes	3-32
Namespace Collision	3-32
Compatibility	3-33
UNIX95 Commands/Libraries	3-38
HP Common Desktop Runtime Environment (CDE 1.0)	3-39
Development Environment	3-40
Printing and Building Help Volumes in Multibyte Locales	3-40
CDE Migration Tool	3-40
Accessing Font Aliases for Remote CDE Sessions	3-41
Changing Languages Between Sessions in CDE	3-42
HP CDE and Commercial Security (Trusted Systems)	3-43
CDE Limitations	3-44
Addition to <i>Advanced User's and System Administrator's Guide</i>	3-44
NIKE Array Enhancements	3-45
60K File Descriptors	3-45
Impacts	3-46
Alternatives	3-47
Size Requirement	3-48
Dynamic Host Configuration Protocol (DHCP)	3-48

Additional Tools . . . . .	3-49
VME Services . . . . .	3-50
Large Buffer Cache . . . . .	3-51
pstat(2) . . . . .	3-51
Remote Procedure Call (RPC) Interfaces . . . . .	3-52
<rpcsvc/*.h> . . . . .	3-52
rpc_control() . . . . .	3-52
RPC 60K File Descriptors . . . . .	3-53
Name Service Switch Support for RPC Lookups . . . . .	3-53
TCP/IP Transport . . . . .	3-53
Cold Install . . . . .	3-53
Software Distributor (SD) . . . . .	3-54
Features . . . . .	3-54
Impact . . . . .	3-55
Obsolete Series 700 Desktop bundles . . . . .	3-55
New CDE Bundles for Series 700 . . . . .	3-56
New CDE Bundles for Series 800 . . . . .	3-56
New Core Bundles for Series 700 . . . . .	3-57
New Core Bundles for Series 800 . . . . .	3-57
Known Problems . . . . .	3-57

#### 4. Major Changes for HP-UX 10.0 and 10.01

New Features of HP-UX 10.0 and 10.01 . . . . .	4-2
Additional 10.01-Specific Changes . . . . .	4-2
Understanding the Changes . . . . .	4-3
Support Information . . . . .	4-4
New Commands . . . . .	4-5
dmpxlt(1) . . . . .	4-5
fstyp(1M) . . . . .	4-6
genxlt(1) . . . . .	4-6
groupadd(1M), groupdel(1M), groupmod(1M) . . . . .	4-6
listusers(1) . . . . .	4-6
logins(1M) . . . . .	4-6
mk_kernel(1M) . . . . .	4-6
model(1) . . . . .	4-7
mountall(1M), umountall(1M) . . . . .	4-7
pwconv(1M) . . . . .	4-7
setuname(1M) . . . . .	4-7

setup(1M) . . . . .	4-7
useradd(1M), userdel(1M), usermod(1M) . . . . .	4-8
vxdump(1M) . . . . .	4-9
vxrestore(1M) . . . . .	4-9
whois(1) . . . . .	4-9
The HP-UX 10.0 File System Layout . . . . .	4-9
Major Changes . . . . .	4-10
File and Directory Locations . . . . .	4-10
System Startup and Shutdown . . . . .	4-15
How This Affects Your System . . . . .	4-15
Using 9.x Pathnames for Files in /etc . . . . .	4-16
Run Levels . . . . .	4-16
Interoperating with 9.x Systems . . . . .	4-16
What Will Happen During the Upgrade to 10.01 . . . . .	4-17
Preparing for the Upgrade . . . . .	4-17
Administering the 10.0 File System . . . . .	4-17
NFS Diskless . . . . .	4-18
Other Documents . . . . .	4-18
What Is NFS Diskless? . . . . .	4-18
Reasons for the Change . . . . .	4-19
New Features and Capabilities . . . . .	4-19
New Parameter . . . . .	4-20
Old (“DUX”) Features No Longer Supported . . . . .	4-20
“DUX” versus NFS Diskless: Summary of Key Differences . . . . .	4-22
Details of Changes for NFS Diskless . . . . .	4-26
New Commands . . . . .	4-26
Changes to Commands and Files . . . . .	4-26
Commands, Parameters, Options No Longer Supported . . . . .	4-27
Upgrading a Cluster to 10.01 . . . . .	4-29
A Useful Script . . . . .	4-29
I/O Convergence . . . . .	4-32
Major Changes for I/O Convergence . . . . .	4-32
Major and Minor Numbers . . . . .	4-32
LU Numbers and Instance Numbers . . . . .	4-33
Interface versus LU . . . . .	4-34
Interface Classes . . . . .	4-34
Guidelines for Working with 10.x Device Files . . . . .	4-35
How This Affects Your System . . . . .	4-35

I/O Convergence and 9.x Code and Scripts . . . . .	4-35
Avoid mknod . . . . .	4-36
Device File Naming Convention . . . . .	4-36
Examples . . . . .	4-38
System Default File Names . . . . .	4-38
How This Affects Your System . . . . .	4-38
Creating Device Files . . . . .	4-38
Disk Management . . . . .	4-39
New Features and Capabilities of LVM . . . . .	4-40
LVM Commands . . . . .	4-41
Products and Features No Longer Supported . . . . .	4-42
Support Matrix for Disk Management Methods . . . . .	4-43
Other Changes . . . . .	4-44
How This Affects Your System . . . . .	4-44
Disk Configurations in 9.x and 10.01 . . . . .	4-44
Drivers . . . . .	4-47
Checklist for Driver Developers . . . . .	4-48
Unmodified Drivers . . . . .	4-48
New Drivers . . . . .	4-49
Driver Name Changes . . . . .	4-49
Drivers Removed at 10.0 . . . . .	4-50
Autochanger Driver and Device Files . . . . .	4-51
Autochanger Virtual Mounting . . . . .	4-51
Series 700 SCSI Subsystem and Disk Driver . . . . .	4-52
Series 700/800 Tape Driver . . . . .	4-53
Tape Driver Differences on the Series 700 and 800 . . . . .	4-54
How This Affects Your System . . . . .	4-54
Drivers May Be Removed . . . . .	4-54
Preparing for I/O Convergence . . . . .	4-55
Kernel Convergence . . . . .	4-56
Major Changes for Kernel Convergence . . . . .	4-56
How This Affects Your System . . . . .	4-57
Preparing for Kernel Convergence . . . . .	4-57
Software Distributor . . . . .	4-59
Reasons for the Change . . . . .	4-59
10.01-Specific Changes to SD . . . . .	4-59
Overview of SD Commands . . . . .	4-60
SD Interfaces . . . . .	4-62

Where to Find Additional Information . . . . .	4-62
Bundles . . . . .	4-62
New Bundle Changes, Tags, and Definitions for 10.01 . . . . .	4-62
10.0 Core Bundles Moving to 10.01 Applications Media . . . . .	4-65
ASE (Asian System Environment) Delivery Re-structuring . . . . .	4-65
Known Problems and Workarounds (for 10.01) . . . . .	4-68
hpterm . . . . .	4-68
fbackup . . . . .	4-68
EISA PSI Card ID . . . . .	4-69
swapon . . . . .	4-69
Unresponsive Small-Memory System . . . . .	4-69
Watchdog Timer . . . . .	4-70
swremove . . . . .	4-70
Kermit . . . . .	4-71
uucp . . . . .	4-71
uucico . . . . .	4-71
uugetty . . . . .	4-71
hpux(1M) . . . . .	4-72
ftp(1)/ftpd(1M) . . . . .	4-72
MPower . . . . .	4-73
SLIP . . . . .	4-73
extendfs(1M) . . . . .	4-73

**5. Other Operating System and Subsystem Changes**

Address Space Layout . . . . .	5-2
Commercial Security . . . . .	5-2
New Features . . . . .	5-2
SVR4: getspent(3C) . . . . .	5-3
New Commands, Calls and Files . . . . .	5-3
Old Features No Longer Supported . . . . .	5-4
Commands, Calls, and Files Changed . . . . .	5-4
Password and Password Aging . . . . .	5-5
Differences Between 9.x and 10.01 . . . . .	5-6
Limitations . . . . .	5-7
Curses . . . . .	5-7
Curses-Color . . . . .	5-9
Effect on Applications . . . . .	5-10
Curses Documentation . . . . .	5-10

DCE (Distributed Computing Environment) . . . . .	5-11
Functionality . . . . .	5-12
Impact . . . . .	5-12
Compatibility . . . . .	5-13
Default-Bundled Software . . . . .	5-15
Optional-Bundled Software . . . . .	5-16
Diagnostics . . . . .	5-17
Recovery . . . . .	5-17
Recovery . . . . .	5-18
Dynamic Buffer Cache . . . . .	5-19
Fast Symlinks . . . . .	5-20
File System Changes . . . . .	5-20
Rock Ridge . . . . .	5-20
Journaled File System . . . . .	5-21
Multiple File System Types . . . . .	5-22
Graphics . . . . .	5-23
HP Process Resource Manager (HP PRM) . . . . .	5-24
HP Remote Watch and HP Support Watch . . . . .	5-25
HP Remote Watch . . . . .	5-25
HP Support Watch . . . . .	5-26
Who To Contact about HP Remote Watch and HP Support Watch . . . . .	5-26
Instant Ignition . . . . .	5-26
Kermit . . . . .	5-28
Kernel Configuration . . . . .	5-29
Kernel Tunables . . . . .	5-29
New Kernel Tunable (o_sync_is_o_dsync) . . . . .	5-29
MC/ServiceGuard . . . . .	5-31
Measurement Systems (KI/MS) . . . . .	5-31
Memory . . . . .	5-31
Virtual Memory . . . . .	5-32
Virtual Memory Policies . . . . .	5-32
Memory Page Deallocation (MPD) . . . . .	5-33
Memory Mapped Files and Semaphores . . . . .	5-34
Multimedia for 10.20 . . . . .	5-35
Capturing Screen Images . . . . .	5-35
Viewing, Editing, and Printing Images . . . . .	5-35
Viewing Images . . . . .	5-36

Printing an Image File . . . . .	5-36
Playing, Recording, and Editing Audio . . . . .	5-36
Supported Workstations and X Stations . . . . .	5-37
Setting Up Audio . . . . .	5-37
Using the Audio Control Panel . . . . .	5-37
Starting the Audio Editor . . . . .	5-37
Using Multimedia at the Command Line . . . . .	5-38
Multimedia for 10.10 . . . . .	5-39
Capturing Screen Images . . . . .	5-39
Viewing, Editing, and Printing Images . . . . .	5-39
Viewing Images . . . . .	5-40
Printing an Image File . . . . .	5-40
Playing, Recording, and Editing Audio . . . . .	5-40
Supported Workstations and X Stations . . . . .	5-41
Setting Up Audio . . . . .	5-41
Using the Audio Control Panel . . . . .	5-41
Starting the Audio Editor . . . . .	5-41
Changes from Audio on HP-UX 9.0 . . . . .	5-42
Using Multimedia at the Command Line . . . . .	5-42
Peripheral Devices . . . . .	5-43
POSIX Realtime Extensions . . . . .	5-44
Scheduling . . . . .	5-44
ptrace() Enhancements . . . . .	5-45
Synchronized I/O . . . . .	5-45
Programming Languages . . . . .	5-46
libM Convergence . . . . .	5-46
matherr . . . . .	5-46
HUGE_VAL . . . . .	5-47
Library Convergence . . . . .	5-47
Millicode . . . . .	5-47
Functions Moved From libc to libm . . . . .	5-48
Automatic Vector Support in Math Library . . . . .	5-48
Linker and Dynamic Loader . . . . .	5-48
Linker . . . . .	5-48
Dynamic Loader . . . . .	5-49
Impact . . . . .	5-50
Performance . . . . .	5-50
Compatibility . . . . .	5-50

Alternatives . . . . .	5-50
Obsolescence . . . . .	5-50
Fastbind . . . . .	5-51
Large UID Support in the Archiver (AR) . . . . .	5-51
Large Files . . . . .	5-52
Linker On-line Help . . . . .	5-52
Impact . . . . .	5-53
Performance . . . . .	5-53
Compatibility . . . . .	5-53
HP-UX Linker (ld) . . . . .	5-53
Impact . . . . .	5-54
Performance . . . . .	5-54
HP C/ANSI C . . . . .	5-54
Performance . . . . .	5-55
Obsolescence . . . . .	5-56
HP C/HP-UX . . . . .	5-56
HP C++ . . . . .	5-58
HP Micro Focus COBOL/UX . . . . .	5-59
HP FORTRAN/9000 . . . . .	5-59
HP Pascal/HP-UX Release 10.0 . . . . .	5-60
PA-RISC Assembler Release 10.0 . . . . .	5-60
SAM (System Administration Manager) for 10.20 . . . . .	5-61
On-line Help Localization . . . . .	5-61
Functional Area Launcher . . . . .	5-61
Restricted SAM Builder . . . . .	5-61
Trusted Systems . . . . .	5-62
Backup and Recovery . . . . .	5-62
Users and Groups . . . . .	5-62
Printers and Plotters . . . . .	5-62
Terminals and Modems . . . . .	5-63
Performance Monitors . . . . .	5-63
System Properties . . . . .	5-63
Disks and File Systems . . . . .	5-63
UUCP . . . . .	5-64
SAM (System Administration Manager) for 10.10 . . . . .	5-64
SAM (System Administration Manager) for 10.0 . . . . .	5-65
SCSI Pass-Through for Series 800 NIO Bus . . . . .	5-67
SharedPrint/UX . . . . .	5-68



SharedPrint/UX in HP-UX 10.0 Versus HP MPower . . . . .	5-69
Additional Printer Support . . . . .	5-69
User Interface Changes . . . . .	5-69
Setting Up SharedPrint/UX . . . . .	5-70
Configuring a Jet Direct Printer . . . . .	5-70
Using the SharedPrint/UX Dialog . . . . .	5-71
Printing with Default Options . . . . .	5-71
Printing a Screen Image . . . . .	5-72
SharedX . . . . .	5-72
Shells . . . . .	5-73
New Default sh . . . . .	5-73
POSIX Shell . . . . .	5-73
Logging . . . . .	5-73
Bourne Shell . . . . .	5-74
Differences between the Bourne and POSIX Shells . . . . .	5-75
POSIX Shell Features New to Bourne Shell Users . . . . .	5-76
Korn Shell . . . . .	5-79
Differences between the Korn and POSIX Shells . . . . .	5-79
Small Disks and Memory . . . . .	5-83
Small Disk Configuration . . . . .	5-83
Small Memory Configuration . . . . .	5-83
FreeDisk . . . . .	5-84
Network Font Services . . . . .	5-84
System Calls . . . . .	5-85
poll() . . . . .	5-85
pstat() . . . . .	5-85
vfork() . . . . .	5-85
<b>typedef</b> Changes . . . . .	5-86
UUCP . . . . .	5-86
Complies with 10.0 File System Layout . . . . .	5-86
/usr/include/dial.h Changes . . . . .	5-88
Complies with NFS Diskless File Sharing Paradigm . . . . .	5-89
Supported over TCP/IP . . . . .	5-89
Conversion Scripts No Longer Supported . . . . .	5-90
VUE (HP VUE 3.0) . . . . .	5-91
HP VUE 3.0 and MPower Announcement . . . . .	5-91
HP VUE 3.0 and Large User IDs . . . . .	5-91
HP VUE 3.0 Troubleshooting and Miscellaneous Changes . . . . .	5-91

Diagnosing VUE Session Startup Problems . . . . .	5-91
Hostname Aliases and HP VUE . . . . .	5-91
Upgrading Users from VUE 2.01 to VUE 3.0 . . . . .	5-92
Getting VUE Running on Non-Default Displays . . . . .	5-92
X11R4/Motif 1.0/1.1 Application Font Usage . . . . .	5-93
Vuelogin . . . . .	5-94
Vuelogin 'langSetup' Resource . . . . .	5-94
Support for New Locales in VUE . . . . .	5-95
X11 New Bitmap Fonts . . . . .	5-95
X11 FontServer for 10.20 . . . . .	5-95
TrueType Fonts . . . . .	5-96
Administering Charsets for True Type Fonts . . . . .	5-96
Wildcard Aliases . . . . .	5-96
X11 FontServer for 10.10 . . . . .	5-96
Font Manipulation . . . . .	5-97
Charset Encoding Extension . . . . .	5-97
Glyph Caching and Scalable Aliases . . . . .	5-97
New Rasterizer . . . . .	5-97
New Fonts . . . . .	5-98
Command Changes . . . . .	5-98
Image Help When Using Localized Environments with HP VUE . . . . .	5-98
Text Help When Using Roman8 Locales with HP VUE . . . . .	5-98
HP VUE and Commercial Security (Trusted Systems) . . . . .	5-99
Changes in Fileset Organization . . . . .	5-99
Toggling Word Wrap Mode in Text Editor . . . . .	5-100
Use of /etc/profile During VUE Session Startup . . . . .	5-100
Functions Marked for Obsolescence . . . . .	5-101
Changes in Keyboard Functionality . . . . .	5-102
Rendering International Characters with 10.0 . . . . .	5-102
Terminal Emulators . . . . .	5-104
Font Server: Dynamically Loaded Rasterizers . . . . .	5-105
X Windows . . . . .	5-105
X11 Fonts . . . . .	5-105
Motif, Xlib, and Intrinsic Libraries . . . . .	5-106
X11 Clients . . . . .	5-106
Color Support for X11 Clients . . . . .	5-106
Xterm . . . . .	5-106

X11 Server and X11 Window System . . . . .	5-107
R6-Based X11 Server . . . . .	5-107
HP X Server Information File . . . . .	5-108
X*screens Format . . . . .	5-109
New X*screens File Format . . . . .	5-109
Setting the Default Visual . . . . .	5-109
Setting the Monitor Size . . . . .	5-109
X Server Environment Variables . . . . .	5-110
Supported Graphics Devices . . . . .	5-111
Double Buffer Extension (DBE) . . . . .	5-112
Multi-Head Support . . . . .	5-113
Multi-Screen Support . . . . .	5-115
Single Logical Screen (SLS) . . . . .	5-115
Shared Memory Extension (MIT-SMT) . . . . .	5-116
Supported Devices . . . . .	5-116
Shared Memory Transport (SMT) . . . . .	5-117
Starting VUE or CDE with the SMT Transport . . . . .	5-119
Possible Extra Files Generated with SMT . . . . .	5-119
X Image Extension (XIE) . . . . .	5-120
Display Power Management Signaling . . . . .	5-120
New Graphics Support . . . . .	5-121
New Dynamic Loading of DDX Drivers and Extensions . . . . .	5-122
Graphics In SAM . . . . .	5-122
Include Inferiors Fix . . . . .	5-122
<b>6. Networking</b>	
BSD Sockets . . . . .	6-2
Data Link Provider Interface (DLPI) . . . . .	6-3
FDDI SNMP Subagent . . . . .	6-4
Impact . . . . .	6-4
Restrictions . . . . .	6-4
Internet Services (ARPA Services) . . . . .	6-4
4-Byte EUC rlogin . . . . .	6-5
Secure Internet Services . . . . .	6-6
System Requirements . . . . .	6-6
Environment . . . . .	6-7
Installing/Enabling . . . . .	6-7
Usage . . . . .	6-8

Compatibility . . . . .	6-8
sendmail for 10.20 . . . . .	6-9
Compatibility and Installation . . . . .	6-11
Performance . . . . .	6-12
sendmail for 10.0 . . . . .	6-12
Variable Length Subnet Masks . . . . .	6-12
Domain Name Server (DNS) Resolver Changes . . . . .	6-13
mrouted (DVMRP) . . . . .	6-15
Name Service Switch . . . . .	6-15
Talk Service ( <code>talk/ntalkd</code> ) . . . . .	6-17
gated . . . . .	6-17
Boot Relay Agent . . . . .	6-19
Trivial File Transfer Protocol Server . . . . .	6-19
Remote Distribution Service ( <code>rdist</code> ) . . . . .	6-20
Network Time Protocol . . . . .	6-20
File Transfer Protocol Service . . . . .	6-21
nslookup . . . . .	6-21
mailx . . . . .	6-22
telnet . . . . .	6-22
whois . . . . .	6-23
DDFA . . . . .	6-23
ARPA Transport . . . . .	6-24
LAN/9000 . . . . .	6-25
48 LAN Card Support . . . . .	6-25
LANCONFIG . . . . .	6-26
LANSCAN/LANADMIN/LINKLOOP . . . . .	6-26
NETTUNE . . . . .	6-26
LAN Product Keywords . . . . .	6-27
LAN Cable Disconnect . . . . .	6-27
NETTRACELOG . . . . .	6-28
Networking <code>_r</code> Routines . . . . .	6-29
Compatibility . . . . .	6-29
Changes Affecting Routines . . . . .	6-30
NFS . . . . .	6-30
NFS Commands . . . . .	6-31
NIS . . . . .	6-32
NS/9000 and vt3k/9000 . . . . .	6-34
ONC/NFS Large UID Support . . . . .	6-34

PMTU . . . . .	6-35
SLIP and CSLIP . . . . .	6-35
SLIP . . . . .	6-35
CSLIP . . . . .	6-36
Sockets . . . . .	6-37
Socket Interfaces . . . . .	6-37
accept . . . . .	6-37
bind . . . . .	6-38
connect . . . . .	6-38
getpeername . . . . .	6-38
getsockname . . . . .	6-38
getsockopt . . . . .	6-38
recv . . . . .	6-39
recvfrom . . . . .	6-39
recvmsg . . . . .	6-39
send . . . . .	6-41
sendmsg . . . . .	6-41
sendto . . . . .	6-41
setsockopt . . . . .	6-41
shutdown . . . . .	6-42
Sockets Header Differences . . . . .	6-42
<fcntl.h> . . . . .	6-42
<sys/socket.h> . . . . .	6-42
<sys/stat.h> . . . . .	6-43
IP Address Resolution Interfaces . . . . .	6-43
gethostent, gethostbyaddr, gethostbyname, sethostent, and endhostent . . . . .	6-43
getnetent, getnetbyaddr, getnetbyname, setnetent, and endnetent . . . . .	6-43
htonl, htons, ntohl, and ntohs . . . . .	6-44
inet_addr, inet_network, inet_makeaddr, inet_lnaof, inet_netof, and inet_ntoa . . . . .	6-44
IP Address Resolution Headers . . . . .	6-44
<arpa/inet.h> . . . . .	6-44
<netdb.h> . . . . .	6-44
<netinet/in.h> . . . . .	6-45
Impact . . . . .	6-45
Compatibility . . . . .	6-45

Size Requirement . . . . .	6-45
Streams . . . . .	6-46
Default MP Synchronization Change . . . . .	6-46
Extended Signals . . . . .	6-46
XTI Internet Protocol-Specific Behavior of Urgent Data . . . . .	6-47
XTI/TLI and TIMOD Use . . . . .	6-47
Streams PTY . . . . .	6-48
L_LIST ioctl . . . . .	6-48
timod for XTI/TLI Libraries . . . . .	6-48
Streams Functionality . . . . .	6-49
STREAMS-Based Pipes . . . . .	6-51
TERMIO . . . . .	6-51
XTI . . . . .	6-52

## 7. Commands and Libraries

Accounting Commands . . . . .	7-2
File System Layout Changes . . . . .	7-2
Changed Command runacct(1M) . . . . .	7-2
Changed Commands in Section 1 . . . . .	7-3
adjust(1) . . . . .	7-6
admin(1) . . . . .	7-7
ar(1) . . . . .	7-7
asa(1) . . . . .	7-7
at(1) . . . . .	7-8
awk(1) . . . . .	7-8
bc(1) . . . . .	7-10
cal(1) . . . . .	7-10
cancel(1) . . . . .	7-10
cat(1) . . . . .	7-11
chown(1), chgrp(1) . . . . .	7-11
cksum(1) . . . . .	7-11
compress (1) . . . . .	7-12
cp(1) . . . . .	7-12
cpio(1) . . . . .	7-13
crontab(1) . . . . .	7-14
csh(1) . . . . .	7-14
ctags(1) . . . . .	7-14
cu(1) . . . . .	7-15

date(1)	7-16
dd(1)	7-16
diff(1)	7-16
doschmod(1), etc.	7-16
du(1)	7-17
echo(1)	7-17
eucset(1)	7-17
ex/vi(1)	7-18
expand(1)	7-21
expr(1)	7-21
find(1)	7-21
findmsg(1)	7-22
Impact	7-23
Performance	7-23
ftio(1)	7-23
Compatibility	7-23
Alternatives	7-23
gencat(1)	7-24
getconf(1)	7-24
grep(1), egrep(1), fgrep(1)	7-24
help(1), sccshelp(1)	7-25
iconv(1) and iconv(3)	7-25
Impact	7-26
id(1)	7-26
iostat(1)	7-27
join(1)	7-27
locale(1)	7-27
ksh(1)	7-27
lex(1)	7-28
lp(1), lpadmin(1)	7-28
lpstat(1)	7-29
ls(1)	7-29
m4(1)	7-30
mail(1)	7-30
make(1)	7-31
man(1)	7-32
mesg(1)	7-32
mkdir(1)	7-32

mkfifo(1)	7-33
mm(1)	7-33
more(1)	7-33
mt(1)	7-34
mv(1)	7-34
nice(1)	7-35
nl(1)	7-35
nljust(1)	7-35
nohup(1)	7-36
nroff(1)	7-36
od(1)	7-36
pack(1)	7-37
patch(1)	7-37
pathchk(1)	7-37
passwd(1)	7-37
paste(1)	7-38
pax(1)	7-38
pr(1)	7-39
prs(1)	7-39
ps(1)	7-40
renice(1)	7-41
rm(1)	7-41
rmdel(1)	7-42
rmdir(1)	7-42
sact(1)	7-43
sar(1)	7-43
sccs(1)	7-43
sed(1)	7-44
sh-posix(1)	7-44
sort(1)	7-45
split(1)	7-46
strings(1)	7-46
strip(1)	7-46
sum(1)	7-47
tar(1)	7-47
Extracting Multiple Archives	7-48
time(1)	7-48
top(1)	7-49



tput(1)	7-49
tr(1)	7-50
tsort(1)	7-51
uncompress(1)	7-51
unexpand(1)	7-51
uniq(1)	7-51
unpack(1)	7-52
uucp(1)	7-52
uuencode(1)	7-54
uulog(1)	7-54
uupick(1)	7-55
uustat(1)	7-55
uuto(1)	7-55
uux(1)	7-56
val(1)	7-56
vi(1)	7-56
Impact	7-57
Compatibility	7-58
wc(1)	7-58
what(1)	7-58
who(1)	7-59
xargs(1)	7-59
zcat(1)	7-60
Changed Commands in Section 1M	7-60
File System Administration Commands	7-61
bdf(1M)	7-62
catman(1M)	7-64
config(1M)	7-65
Input and Output Files	7-65
dcopy(1M)	7-66
df(1M)	7-66
dodisk(1M)	7-67
fbackup(1M)	7-68
ff(1M)	7-68
frecover(1M)	7-69
fsck(1M)	7-69
fsdb(1M)	7-71
fuser(1M)	7-72

insf(1M), lssf(1M), mkssf(1M), rmsf(1M) . . . . .	7-72
ioscan(1M) . . . . .	7-73
labelit(1M) . . . . .	7-74
localedef(1M) . . . . .	7-75
Effect of 10.x Locale Changes on 9.x Code . . . . .	7-76
If 9.x Application Uses Customized 9.x Locales, and Linked Shared . . . . .	7-76
If 9.x Application Uses Standard 9.x Locales, and Linked Shared . . . . .	7-77
If 9.x Application Linked Archived . . . . .	7-77
lsdev(1M) . . . . .	7-78
mkboot(1M) . . . . .	7-78
mkfs(1M) . . . . .	7-79
mount(1M), umount(1M) . . . . .	7-80
Summary of Change . . . . .	7-80
Impact . . . . .	7-83
Performance . . . . .	7-83
Limitations . . . . .	7-83
ncheck(1M) . . . . .	7-85
Performance . . . . .	7-85
Compatibility . . . . .	7-85
ncheck_hfs(1M) . . . . .	7-86
newfs(1M) . . . . .	7-86
Changed Options and Behavior . . . . .	7-87
quot(1M) . . . . .	7-89
quotacheck(1M) . . . . .	7-89
savecore(1M) . . . . .	7-89
Impact . . . . .	7-90
Performance . . . . .	7-90
Size Requirement . . . . .	7-91
New on Both Series 700 and 800 . . . . .	7-91
New on Series 700 . . . . .	7-92
Faster system bootup . . . . .	7-92
Smaller corefile . . . . .	7-92
shutdown(1M) . . . . .	7-93
Changing Existing Scripts . . . . .	7-93
swapon(1M) . . . . .	7-94
sysdef(1M) . . . . .	7-94

syslogd(1M)	7-94
unmount(1M)	7-95
volcopy(1M)	7-95
whodo(1M)	7-95
Commands No Longer Supported	7-97
Library Routines (libc)	7-100
Changes for 10.20	7-100
getrlimit()	7-100
lockf()/fcntl()	7-100
ulimit()	7-101
Changes for 10.10	7-101
UNIX 95 Conformance	7-101
New Functionality for 10.10	7-101
Incompatibilities with Previous Versions	7-103
Additional Changes	7-103
curses(3X)	7-104
Changes for 10.0	7-104
SVID3 and XPG4 Conformance	7-104
XPG4 Base Profile Branding	7-104
Summary of Changes for XPG4	7-105
Changes to libc	7-105
Changes to libM	7-106
Changes to Commands	7-106
Changes to Headers	7-107
How This Affects Your Code	7-108
New libc Routines at 10.0	7-109
lckpddf(3C), ulckpddf(3C)	7-110
libc Changes to Support Threads	7-110
Thread-Safe Routines	7-111
New libc Routines for Locking in Threaded Applications	7-112
libc Routines that Behave Differently in Threaded Applications	7-113
libc Routines Not Supported for Threaded Applications	7-113
The 10.0 File System Layout and libc	7-114
Other Changes to libc Routines	7-114
closelog(3C)	7-115
crypt(3C)	7-115
fnmatch(3C)	7-115

glob(3C), globfree(3C) . . . . .	7-116
nftw(3C) . . . . .	7-116
openlog(3C) . . . . .	7-116
regcomp(3C), regexec(3C), regfree(3C), regerr(3C) . . . . .	7-117
setlocale(3C)/iconv(3C) . . . . .	7-118
wordexp(3C), wordfree(3C) . . . . .	7-118
libc Routines No Longer Supported . . . . .	7-118
Math Library Routines (libm, libM) . . . . .	7-119
New C Math Library Functions . . . . .	7-119
Millicode versions of math library functions . . . . .	7-120
New Locations for Math Libraries . . . . .	7-121
How This Affects Your Code . . . . .	7-122
Changes to Other Commands and Files . . . . .	7-123
tztab(4) . . . . .	7-123
Changes to /etc/disktab (disktab(4)) . . . . .	7-124
How This Affects Your System . . . . .	7-125
Variables To Be Removed from unistd(5) . . . . .	7-126

**Index**

## Tables

---

4-1. 9.x to 10.0 Directory Map . . . . .	4-11
4-2. 9.x Commands versus SD Commands . . . . .	4-61
5-1. HP MPower Manpages . . . . .	5-38
5-2. HP MPower Manpages . . . . .	5-43
6-1. NIS Commands . . . . .	6-33



## Overview of Release Notes for HP-UX 10.20

---

This document contains information about the HP-UX 10.20 release (referred to as “10.20”), and, unless specifically stated otherwise, this information also pertains to HP-UX 10.10, HP-UX 10.01, and HP-UX 10.0 systems. See the following chapters for information about the releases:

- Chapter 2—what is new at 10.20
- Chapter 3—what was new at 10.10
- Chapter 4—what was new at 10.0 and what was added at 10.01

This document covers the following topics:

- Major Changes for HP-UX 10.20.
- Major Changes for HP-UX 10.10.
- Major Changes for HP-UX 10.0 and 10.01.
- Other Operating System and Subsystem Changes.
- Networking.
- Commands and Libraries.

This chapter:

- Describes the release notes.
- Explains the purpose of the release notes.
- Points to other sources of information on the HP-UX 10.20 release.

---

**Note** See the section “System Compatibility” in Chapter 2 in this document *before* you begin planning your upgrade to 10.20. The section briefly describes the steps you must take and also points you to tools and other sources of information you will need.

---

---

## The Release Notes and Release Notes Files

The 10.20 release notes are contained in the following:

- HP Help
- HP LaserROM
- an ASCII file that is part of your core HP-UX 10.20 system  
(`/usr/share/doc/10.20RelNotes`)

You will find additional product-specific release notes files under `/opt productname`. For example, DCE (Distributed Computing Environment) release notes are in the `/opt/dce/newconfig/RelNotes` directory. In addition, if you add a new product to your system, its release notes will be loaded into `/opt productname`.

---

## The Purpose of Release Notes

Release notes provide you with information on new, changed, and removed functionality (features). This particular release notes document explains the major differences between HP-UX 9.0 and HP-UX 10.0. It also describes changes for HP-UX 10.01, 10.10, and 10.20.

Release notes do not completely document all of the features of a release. Instead, release notes contain high-level information and provide pointers to more detailed operating system and product-specific documentation.

Where appropriate, release notes also tell you about changes in the support of products.



---

## Other Sources of Information

In addition to these release notes, you have many other sources of information available to you.

### README Documents

README documents contain information about the installation process that may not appear in the installation manual. Any product may have a README document, so you may have several README documents.

### Online Documents

This document and a number of 10.0, 10.01, 10.10, and 10.20 white papers are available online on 10.20 systems in `/usr/share/doc`. For a listing of online documents, refer to Table 1-5 in the *HP-UX 10.x Documentation Map* online white paper. This white paper is available as a PostScript file (`doc_map.ps`) and as an ASCII file (`doc_map.txt`) in `/usr/share/doc`.

---

#### Note

For 10.10 and 10.20, the manpages are available in LaserROM under the title *HP-UX Reference* and via the `man` command.

For 10.01, the manpages are available in LaserROM under the title *HP-UX Reference*, in hardcopy as the *HP-UX Reference*, and via the `man` command.

---

### HP-UX Upgrade Tools Documents

A subset of the online documents is also provided in the 9.x-based package “HP-UX Upgrade Tools for 9.x to 10.x”. After you install the software onto your 9.x system, you will find the documentation files in the `/etc/newconfig/10RelNotes` directory.

---

## The HP-UX Documentation Set

The HP-UX documentation set describes how to set up and use the basic HP-UX system. It includes information on system administration, networking, Windows, and so on.

The documentation set for the HP-UX 10.x releases is very different from that of pre-10.0 HP-UX releases. To make it easier for you to find the information you need, the organization and content of the manual set has been changed: many manuals have been combined, redundant information has been removed, information has been moved between manuals, more information is available online, and some of the information that was in HP manuals is now in trade manuals. For more information on the 10.20 documentation set, refer to the *HP-UX 10.x Documentation Map* online white paper. The *HP-UX 10.x Documentation Map* is available as a PostScript file (`doc_map.ps`) and as an ASCII file (`doc_map.txt`) in `/usr/share/doc`.

## Major Changes for HP-UX 10.20

---

This chapter describes the following HP-UX 10.20 changes:

- 64-bit Data Register
- CDE (HP Common Desktop Runtime Environment)
  - Pluggable Authentication Modules (PAM)
  - HP CDE and Single Logical Screen (SLS) Server
- Cold Install
- DHCP (Dynamic Host Configuration Protocol)
- HP Distributed Print Service (HPDPS)
- HP-UX Bootstrap
- JFS Version 3
- JFS as the Root File System
- Large Files
- Large User IDs
- Multi-Processor Control
- QIC Media
- Reverse ARP (Address Resolution Protocol)
- SD-UX
- System Compatibility
- VM Kernel Address Space
- VME Services

Also see the following chapters for additional 10.20 changes:

- **Chapter 5**, “Other Operating System and Subsystem Changes”:
  - DCE (Distributed Computing Environment)
  - Multimedia for 10.20
  - Programming Languages
  - SAM (System Administration Manager)
  - SharedPrint/UX
  - SharedX

- `typedef` Changes
- VUE (HP VUE 3.0)
- X Windows
- **Chapter 6, “Networking”:**
  - Data Link Provider Interface (DLPI)
  - FDDI SNMP Subagent
  - Internet Services
  - Networking `-r` Routines
  - NIS
  - ONC/NFS Large UID Support
  - Secure Internet Services
  - `sendmail`
- **Chapter 7, “Commands and Libraries”:**
  - Commands in Section 1
  - Commands in Section 1M
  - Library Routines (`libc`)

---

## 64-bit Data Register

For 10.20, the compilers provide the capabilities to build with specialized instructions to take advantage of performance features on PA2.0 systems. The `+DA2.0` flag provides these features and is included in the default flags on PA2.0 systems. One of the features provided by the compiler is the ability to use 64-bit registers for arithmetic operations to enhance the performance of the compiled code.

### Summary of Change

Only machines with the PA8000 processor chip are affected.

Code generated with the PA2.0 compiler flag set will use 64-bit data registers, where appropriate, for arithmetic operations. The HP-UX operating system has been modified to save and restore 64 bits worth of register information on PA2.0 systems. Applications that implicitly make use of register information returned by the operating system via the `sigcontext` structure will have to be

## 2-2 Major Changes for HP-UX 10.20

addressed to handle 64-bit data on PA2.0 systems, unless they include no code that has been generated with the +DA2.0 compiler flag.

## **Impact**

Any application that links in any code compiled with the +DA2.0 compiler flag, including any object modules, shared or archived libraries, and that has signal handlers that access the register contents in the sigcontext structure must be prepared to handle 64-bit data registers unless the code that is compiled with the +DA2.0 flag blocks signals. If this is not done, it is possible that either program termination or data corruption may occur, if for example, the application restores only 32-bits of a 64-bit data register and returns to a section of code that needs the 64-bit data. Note that this interface of accessing the machines registers is not part of the HP-UX API and its use is not encouraged because the information is strictly machine-dependent and cannot always be made compatible as modifications are made to the underlying machine architecture.

In addition, code that does access the register contents in the sigcontext structure, will fail the compile process on a HP-UX 10.20 release until it has been re-coded to match new names for these registers in the sigcontext structure. This data was purposefully renamed to alert users of the coding changes that may be necessary to account for the larger data.

As this is not a documented interface, the source code modifications that will need to be made can be found along with the structure definition in the `save_state.h` header file.

## **Performance**

The 64-bit data registers feature of the PA2.0 compilers is meant to increase the performance of the applications that can make use of 64-bit arithmetic. Performance benefits will be different depending on how much the application may be able to take advantage of this feature.

## Compatibility

Applications that are impacted by this feature may abnormally terminate or result in data corruption until source code changes are made to the part of the application that accesses the register information in the savestate portion of the sigcontext structure. In addition, if the code that accesses the registers is recompiled on HP-UX 10.20, it will fail the compile until source code modifications have been made.

Note there is no compatibility issues with this feature if the +DA2.0 compile flag is not used on any of the code contained in the application, or if the only code compiled with the +DA2.0 flag has signals blocked, or if the application does not access the register contents of the sigcontext structure.

As this is not a documented interface, the source code modifications that will need to be made can be found along with the structure definition in the `save_state.h` header file.

---

## CDE (HP Common Desktop Runtime Environment)

For full details on CDE, refer to the section “HP Common Desktop Runtime Environment (CDE 1.0)” in Chapter 3. This section only discusses topics that are relevant for 10.20.

### Pluggable Authentication Modules (PAM)

PAM is the new industry standard integrated login framework.

The PAM framework is used by system entry components, such as `dtlogin`, to authenticate users logging into the system.

In HP-UX 10.20, PAM is introduced to be used only by CDE authenticating components. Currently, non-CDE authenticating components, such as `login` and `ftp`, continue to use the HP proprietary integrated login framework. These authenticating components will be gradually modified to use PAM in future releases, thus obsoleting HP ILOGIN framework.

The PAM framework is used for easy integration of additional security technologies into HP-UX system entry commands, such as `dtlogin`. The

## 2-4 Major Changes for HP-UX 10.20

security technologies being integrated for HP-UX 10.20 are generic HP-UX (`/etc/passwd`), Commercial Security, and DCE.

For 10.20, CDE components (`dtlogin`, `dtsession`, and `dtaction`) will use PAM to authenticate users, as well as to establish user credentials (for example, for DCE).

For 10.20, CDE components are also capable of authenticating users using the commercial security databases.

### **Impact**

The CDE users on systems belonging to DCE cells will be able to authenticate themselves with the DCE registry and obtain DCE credentials at the login time.

System administrators can require CDE users to conform to the security policies enforced in the trusted system databases.

### **Configuration**

A new configuration file `/etc/pam.conf` is used to determine usage of security mechanisms to authenticate users. The presence of PAM and its configuration file will not be noticed on systems configured to use generic HP-UX-based user authentication (`/etc/passwd`) or commercial-security based user authentication. For those systems intending to use DCE integrated login functionality within CDE, the `auth.adm` utility will create the desired configuration file that is functionally equivalent to the corresponding ILOGIN `auth.conf` file created for HP-proprietary integrated login framework.

Refer to the `pam.conf(4)` and `pam(3)` manpages for additional information.

### **Known Problems**

#### **Vacation Mail**

The HP-UX vacation program does not support the `$$SUBJECT` macro. Vacation mail functionality is described in Chapter 8 of the *CDE User's Guide* under the topic "To Send an Automatic Message (Vacation Mail)".

The known problem in 10.10 of changing word wrap mode is not applicable to 10.20.

## Error Messages

The 10.20 CDE desktop does not contain updated versions of SharedPrint or VUE. Therefore, when updating from the VUE desktop to the CDE desktop, existing operating system versions of SharedPrint or VUE remain on the system. This results in SharedPrint and VUE errors appearing in `swagent.log` and possibly `swverify.log`.

The following filesets will produce a failed verify error message:

```
SharedPrint.SHP-ENG-A-MAN
SharedPrint.SHPRNT-CLI
SharedPrint.SHPRNT-PCL
SharedPrint.SHPRNT-SRV
```

```
VUE.VUE-RUN
VUE.VUE-HELP-INFO
VUE.VUE-MAN
VUE.VUE-RUN-AUX
```

(Non-English systems will see the localized fileset names.)

These error messages can be ignored. To ensure binary compatibility, your 10.01 and 10.10 files will work on HP-UX 10.20.

## HP CDE and Single Logical Screen (SLS) Server

For 10.20, HP CDE will have built-in support for the Single Logical Screen (SLS) server. The implication of this change is that on an SLS-enabled server, dialogs displayed by CDE clients that center themselves on the screen will instead center themselves on a particular monitor as opposed to the center of the “screen”.

The CDE clients have been changed to understand SLS configuration. How clients respond to x,y resources also changed. By default, the clients will put the windows on one physical screen. If this is not desired, the resources are available to change this behavior. Changed CDE clients are:

- `dtgreet` - login screen and sub-windows
- `dtwm` - initial front panel placement, move feedback window, other dialogs
- `dthello` - placement of copyright text

### 2-6 Major Changes for HP-UX 10.20



- dtsession - logout confirmation dialog, screen lock
- libDtSvc - action dialogs, errors, prompts

To determine whether a display is an SLS display, the same method used by the server to determine whether a display was set up for SLS is used. That is, examine the property for SLS on the root window to see if it is configured. An API call is set up in DtSvc which passes back the number of rows and columns supported in the SLS configuration. If these values return 0, the display is not an SLS display. The function definition is:

```
void
parseXnScreensFile(slsRows, slsColumns)
int *slsRows;
int *slsColumns;
```

Now that there is a function that determines whether a display is in SLS mode, the correct clients are changed to use this function and perform the appropriate action if running on an SLS display. The best way to determine which monitor to center on was by way of resources. What was done for each client that needed to 'center' dialogs was to define two new resources for each client:

```
singleLogicalScreenX:
singleLogicalScreenY:
```

The clients that had these resources added to are:

```
dtgreet
dtwm
dtsession
```

The logic for these two resources follows the 'X' standard mechanism for assigning its' coordinates (that is, x,y where 0,0 is the Upper Left hand corner of the screen). For example, if an SLS display is configured with four monitors such that it is layed out 2 by 2 and the user specifies

```
*singleLogicalScreenX: 2
*singleLogicalScreenY: 1
```

the clients would center its dialogs on the upper right-hand monitor. The layout for the 2 by 2 setup would be:

```
1,1 2,1
1,2 2,2
```

To center the Front Panel, the logic used is to center on the '\*singleLogicalScreenX:' location at the bottom of all the monitors.

Because the Login Manager uses Xresources to define its' resources, to set up a CDE session for SLS, you must modify these two files:

```
/etc/dt/config/<LANG>/Xresources
/usr/dt/app-defaults/<LANG>/Dt
```

Now there is a mechanism to determine whether a display is an SLS display (parseXnScreensFile()) and has resources to support placement of centering of dialog, code changes were made in clients to force the dialogs to the correct location. The clients modified are:

```
dtgreet
dthello
dtwm
dtsession
```

---

## Cold Install

The new 10.20 features added for the system installation process are:

- **VxFS**—The root LVM volume (/) can be configured with the VxFS file system. This is the default when choosing the LVM with the VxFS option.
- **DHCP**—Support for obtaining and allocating IP address and networking information via the DHCP (Dynamic Host Configuration Protocol). If a DHCP server is available, systems doing installs will contact the server to get a default set of network information.
- **Locale**—The system's language locale will be set depending on which language bundle is chosen during the installation.
- **Striping**—Using LVM, you can choose to configure multiple disks in same volume group with data striping for increased performance.
- **Keywords**—Several new keywords are added to the configuration file syntax for setting the keyboard language (`kbdlang`), SD secrets file (`sd_secrets`), and hook for running a script prior to running SD (`pre_sd_cmd`). See the `instl_adm(4)` manpage for details.

### 2-8 Major Changes for HP-UX 10.20

---

## DHCP (Dynamic Host Configuration Protocol)

For 10.20, `dhcpcclient` can be called to get a DHCP lease from a DHCP server (there is a DHCP server on HP-UX starting with 10.10 and other vendors also produce DHCP servers). This information includes an IP address and subnet mask that the machine should use, as well as other optional information, such as a DNS server and/or NIS server. DHCP is defined in RFC's 1541, 1542, 1533, and 1534; these definitions have been updated in later versions of the standards documents.

`dhcpcclient` is only intended to run if it is invoked by `auto_parms` or by Cold Install (that is, it is not intended to be run directly). It will not be invoked on diskless systems, but will be invoked on systems where `DHCP_ENABLE[0]` is set to 1 in `/etc/rc.config.d/netconf`. Cold Install will offer you the address that `dhcpcclient` gets from a server, but if you reject it, Cold Install will tell `dhcpcclient` to tell the server that the client does not want the address. Addresses can be given to the client permanently or they can be leased for a finite time by the server. The client will be invoked by `auto_parms` at system start up and the lease will be extended if appropriate (`dhcpcclient` might also be asked to keep extending the lease as needed while the system remains up).

### Impact

Currently, Cold Install and Instant Ignition require the machine's user to manage their own IP address. While you can still manually manage your own IP address, you can now allow Cold Install or Instant Ignition to query a DHCP server to get your IP address so that the IP address can be centrally managed on a server. Other configuration information (such as hostname and various server addresses) can also be obtained from the server in the same way.

With the introduction of DHCP support in both the system installation tools and in the `set_parms` utility (via `auto_parms`), new systems can be easily added to a network. DHCP allows for dynamic IP address, hostname allocation, and IP lease management, as well as supplying general network information such as default route, netmask, NIS domain, DNS server, and so on.

## Performance

Performance is only potentially impacted at system start up. If Cold Install or `auto_parms` calls `dhcpcclient`, `dhcpcclient` will attempt to contact a DHCP server. This does not usually take long if there is a server, but it can take long if the network is busy. However, if there is no server, the process will be slowed. In this case, `dhcpcclient` will take time to timeout. When `dhcpcclient` is invoked, the amount of time that `dhcpcclient` takes between retries and how many times it retries to contact a server can be specified on the command line.

## Alternatives

If `DHCP_ENABLE[0]` is set to 0 in `/etc/rc.config.d/netconf`, the client should not be invoked by `auto_parms`. If running Cold Install, you can choose to *not* use the address offered by Cold Install via `dhcpcclient`. Cold Install will then set `DHCP_ENABLE[0]` to 0 and tell `dhcpcclient` to give the address back to the DHCP server.

## Configuration

If `DHCP_ENABLE[0]` is set to 0 in `/etc/rc.config.d/netconf`, the client should not be invoked. If `DHCP_ENABLE[0]` is set to 1, `dhcpcclient` should be invoked at system start up.

---

## HP Distributed Print Service (HPDPS)

The HP Distributed Print Service (HPDPS) provides a standards-based print spooling service for the networked enterprise. The HPDPS is based on standards ISO DPA 10175 and POSIX 1387.4, which are in turn based on Palladium Version 2 technology from MIT's Project Athena.

The HPDPS provides:

- Support for the large enterprise. The HPDPS can handle networks with thousands of printers, spoolers, and clients, yet avoids impairing performance in large configurations of old spooler systems.

### 2-10 Major Changes for HP-UX 10.20

- Access from the HP VUE and CDE window managers. Printer icons appear on the desktop for HPDPS printers created on your host or within your DCE cell. All window manager print system operations can be performed on the HPDPS.
- Optional use of HP Distributed Computing Environment (DCE) cell services. When executed in a DCE cell, the HPDPS provides these features in conjunction with DCE:
  - Single-point administration of the HPDPS from any host in your cell. For example, a new printer is created once and the entire cell can instantly access the new printer.
  - Security authentication and authorization for all access to the HPDPS using DCE Access Control Lists to control access. For example, certain users can be restricted from printing to a certain printer or from printing jobs larger than a certain size.
- Automatic selection of printers appropriate for a print job. For example, the HPDPS can automatically select a printer capable of double-sided printing for a job that requires it. Conversely, with the HPDPS, you can disallow submission of jobs based on DCE authorization, job size, paper selection, and other attributes.
- Ability to configure pools of printers to distribute the workload among several printers. Queued jobs are routed to the next available printer from the pool, rather than queuing up several jobs for the same printer. Thus, the print load is balanced.
- Monitoring of printer status during the printing of a job using SNMP (for printers with SNMP support). The HPDPS detects conditions such as low toner or paper jam.
- User and operator notification of events such as print system problems and changes in job status. For example, the HPDPS can be configured to send email to the operator when a printer runs out of paper or when the number of print jobs in a queue becomes too large.
- Convenient administration of the HPDPS using SAM. With SAM, you can create, modify, delete, and view the status of HPDPS objects.
- Interoperability with existing `lp-` and `lpr-`based spoolers on supported platforms, including HP-UX hosts running the LP spooler. A print job

submitted using the LP spooler on one host can be transferred to another host for processing by the HPDPS, and vice-versa.

- Access to the HPDPS from the LP spooler commands. Using the familiar LP spooler commands, such as `lp`, `lpstat`, and `cancel`, print jobs can be submitted, listed, and canceled within the HPDPS. Existing software that executes the LP spooler commands for these operations can now access the HPDPS as well.
- Support for a variety of printer and plotter devices. HPDPS supports an extensive list of HP printers and plotters, including the LaserJet 5, Color LaserJet 5, DeskJet 1600CM, DeskJet 1200C/PS, and DesignJet 650C. Support for a new printer or plotter model can be added to a running HPDPS server without shutting down the server.

## Impact

Both the LP spooler and the HPDPS can be installed, configured, and executed on the same host. If you choose not to use the HPDPS, you will see no effect on performance or memory usage.

## Size Requirement

See the *HP Distributed Print Services Administration Guide* for information on memory and disk requirements.

## Additional Information

For details on the HPDPS, see the *HP Distributed Print Service Administration Guide*, the *HP Distributed Print Service User's Guide*, and the `pdpr(1)` and `pd_att(5)` manpages.

---

## HP-UX Bootstrap

For 10.20, `hpux(1M)` supports the loading of large kernels. This is accomplished by having `hpux(1M)` move itself to a higher position in memory to allow more room for the kernel being loaded.

In addition, `hpux(1M)` supports separate boot and root volumes with LVM. This allows the system to have a small boot volume and a large root volume.

Existing systems do not need to change configuration.

---

## JFS Version 3

For 10.20, the kernel and the JFS commands now support a new disk layout for JFS file systems called Version 3. The existing Version 2 layout is also supported. This means that file systems of either type can be mounted and all JFS commands work on both versions.

By default, `mkfs(1M)` creates a Version 3 layout, although there is an option to create a Version 2 layout. There is a new command, `vxupgrade(1M)`, to convert Version 2 file systems to Version 3. The upgrade is performed on-line (that is, while the file system is mounted). You do not have to reboot the system, change any configuration files (such as `/etc/fstab`), or even unmount the file system. Once a JFS file system has been upgraded to the Version 3 disk layout, the process cannot be reversed.

JFS file systems created during a cold installation will use the new Version 3 layout. However, if an earlier release is updated to HP-UX 10.20, the JFS file systems will still use the Version 2 layout. You can choose to upgrade the pre-existing file systems to Version 3 using the `vxupgrade(1M)` command.

Note that a Version 3 JFS file system cannot be mounted by an earlier release of HP-UX. Therefore, under some circumstances, you might prefer to create a new JFS file system with the older Version 2 layout.

Several features of HP-UX 10.20 are supported on JFS only with the new Version 3 layout. These include large files and large UIDs (see the sections “Large Files” and “Large User IDs” in this chapter for details). It is not

possible to create large files or files owned by users with large UIDs on a Version 2 JFS file system.

HP-UX 10.20 also supports the new Data Management API defined by the Data Management Interface Group (DMIG), but only on JFS file systems using the Version 3 disk layout. The DMAPI is not supported on Version 2 JFS file systems or on HFS file systems.

### **Impact**

The JFS Version 3 disk layout will not affect any existing applications. The kernel and commands continue to support all JFS interfaces that were supported in past releases of HP-UX. Applications should not rely on the file system's internal data structures. If they do, applications might not work correctly.

### **Compatibility**

For 10.20, JFS is source-compatible and binary-compatible with earlier releases of HP-UX. This is true whether the new Version 3 disk layout or the older Version 2 disk layout is used.

### **Alternatives**

As in the past, you can create either HFS or JFS file systems and can freely mix the two on a single computer. If you create JFS file systems, you now have the choice of using the Version 2 or the Version 3 disk layout. These versions can also be freely mixed.



---

## JFS as the Root File System

For 10.20, HP-UX now supports local root file systems of types HFS and JFS (Veritas VxFS). In prior releases, a local root file system had to be type HFS.

File systems in which the local root file system is a whole-disk file system are not affected. These file systems remain type HFS. File systems used for NFS Diskless clusters, in which the root file system is an NFS mount to a file system tree on a server, are not affected.

### **/stand Directory**

The major change to HP-UX system architecture is that the directory `/stand` will be created as a separately mounted file system. `/stand` will always reside on the first logical volume on the bootdisk.

The `/stand` directory, first implemented in a previous 10.0 release, contains the system kernel and I/O mapping files that are required at boot time. In addition, the directory also contains kernel configuration and generation files.

If the directory `/stand` is a separate file system, it must be of type HFS. The root file system (directory “/”) can be of type HFS or JFS. Also, if `/stand` is a separate file system, an entry corresponding to this file system must be maintained in the file `/etc/fstab`. This entry is created when HP-UX is installed and must be maintained. Removing this entry might result in a non-functional system.

The directory `/stand` is now explicitly mounted in user space at startup time, with the mount operation performed in the startup script `/sbin/ioinitrc`. Because `ioinitrc` is the first operation performed from entries in `/etc/inittab`, this is done at the beginning of user space initialization.

`/stand/rootconf` will be used for recovering the system, so you should not delete or modify this file.

## Rebooting

To reboot a system after replacing the kernel, the kernel should be placed in the `/stand` directory only. If the kernel is placed under `/`, the secondary loader `hpux(1M)` will not be able to read the new kernel because `/stand` and `/` are different file systems.

## Mirroring

If the boot volume and the root volume need to be mirrored, the boot volume should be mirrored first. That is, the `lvextend(1M)` command should be used for the boot volume before it is used for the root volume. Both the boot and the root volume need to be mirrored because mirroring only one volume will not allow the system to boot from the mirror.

## Boot Messages

During the maintenance mode boot, the console will display three messages stating that `/stand` could not be mounted and the logical volume used for `/stand` could not be found. These messages can be ignored.

## System Recovery

During an HP-UX system recovery, the following files may be needed to be restored from the backup.

```
/stand/rootconf
/etc/lvmtab
/etc/fstab
/stand/ioconfig
```

These files should be backed up using any of the following utilities: `tar`, `cpio`, or `fbackup`. If you do not have a backup copy of these files, your system may be unrecoverable.

## Local Root File Systems

For local root file systems, changes were made to `hpux(1M)` (which is the Secondary System Loader (SSL)) and kernel and user-space startup software to support the features described above. At boot time, `hpux(1)` looks for a bootable kernel on HFS volumes in one of these directories:

- `/stand/<kernel_name>`—for the case of an HFS root and no separately mounted `/stand` file system
- `/<kernel_name>`—for the case of a separately mounted `/stand` file system (for HFS root, as well as JFS root)

## User-Space Startup Software

User-space startup software has been changed to remove implicit assumptions that the root file system is of type HFS and also to explicitly mount the `/stand` directory. All startup checks for file system consistency (`fsck` and `fsckclean` checks) have been retained. However, these tests have been modified to first determine the type of file system and then to invoke the correct utility appropriate for that type of file system. Startup programs and scripts are now independent of the file system type of the root file system.

## Kernel Support

Kernel support of JFS file systems was present in earlier releases; the kernel now also supports JFS file systems as the root file system.

## Compatibility

User software should be completely independent of this functionality. All HP-UX system administration tools have been internally modified to support this functionality. You can continue to operate with whole-disk or HFS-only file systems as before.

---

## Large Files

For 10.20, HP-UX now supports files up to 128GB in size. The prior limit was 2GB.

This section summarizes the large files features. For a complete description of the HP large file implementation, see the *HP-UX 10.20 Large Files White Paper* in `/usr/share/doc` (see the section “Online Documents” in Chapter 1 for the exact file name and content description).

Many UNIX vendors support a 32-bit operating system, which inherently has a limitation of 2GB when representing file sizes and offsets. The Large Files Summit, an independent working group consisting of many UNIX vendors, has addressed this limitation and agreed on an interface that enables a 32-bit operating system to support file sizes greater than 2GB. For details, see the website <http://www.sas.com/standards/large.file>. The HP-UX implementation of large files follows the Summit specifications.

Large files (files greater than 2GB) are supported through a new set of system calls and `libc` routines, as well as through HP-UX system administration and user commands. The implementation is supported on UFS and JFS Version 3 file systems, but not on NFS. In addition, the implementation is completely binary compatible with prior releases of HP-UX.

### Interfaces Supporting Large Files

There are two ways to support large files:

1. There is a new set of 64-bit interfaces and data types (similar to 32-bit interfaces).
2. There is a new compile environment that is requested with a new flag in which existing interfaces and data types use 64-bit values to support large files. If an application does not have to interact with large files, there are no required changes. However, the application developer needs to take specific actions to enable large files.

### Data Types

These are the data types that are 64 bits in the alternate compile environment:

`fsblkcnt_t` (new data type)

## 2-18 Major Changes for HP-UX 10.20

```
fsfilcnt_t (new data type)
fpos_t
off_t
rlim_t
```

These data types remain 32 bits in the default environment, although the data types also have 64-bit equivalents in the 32-bit environment. The names end in “64” or “64\_t”, such as `fsblkcnt_t` and `fsblkcnt64_t`.

### Data Structures

Below are the data structures that use the above data types.

```
dinode    off64_t di_size

flock     off_t l_start
          off_t l_len

ic_common off64_t ic_size

inode     off64_t i_size

rlimit    rlim_t rlim_cur
          rlim_t rlim_max

stat      off_t st_size
          fsblkcnt_t st_blocks

statvfs   fsblkcnt_t f_blocks
          fsblkcnt_t f_bfree
          fsblkcnt_t f_bavail
          fsblkcnt_t f_files
          fsblkcnt_t f_ffree
          fsblkcnt_t f_favail
```

In the alternate compile environment, the structures contain the 64-bit fields listed. In the default environment, these structures do not change, but there are parallel structures, such as `flock64` for `flock`.

## Subroutine Interfaces

Below are the subroutine interfaces that are different in the alternate compile environment, usually because they use one of the types or structures above. The name of the corresponding routine usable in the default environment is also listed:

Existing Interface =====	New Interface =====
creat()	creat64()
fgetpos()	fgetpos64()
fopen()	fopen64()
freopen()	freopen64()
fsetpos()	fsetpos64()
fstat()	fstat64()
fstatvfs()	fstatvfs64()
fstatvfsdev()	fstatvfsdev64()
ftruncate()	ftruncate64()
ftw()	ftw64()
getrlimit()	getrlimit64()
lockf()	lockf64()
lseek()	lseek64()
lstat()	lstat64()
mmap()	mmap64()
nftw()	nftw64()
nftw2()	nftw2_64()
open()	open64()
prealloc()	prealloc64()
setrlimit()	setrlimit64()
stat()	stat64()
statvfs()	statvfs64()
statvfsdev()	statvfsdev64()
tmpfile()	tmpfile64()
truncate()	truncate64()

## New Interfaces

New interfaces have been introduced for `fseek()` and `ftell()` because the POSIX definitions do not support file sizes larger than 2GB on machines that use 32-bit `int` values.

## 2-20 Major Changes for HP-UX 10.20

New 32-bit interfaces	New 64-bit interfaces
=====	=====
ftello()	ftello64()
fseeko()	fseeko64()

## Commands

The following commands have also been enhanced to provide command-line support for large files:

- Accounting and `quota` commands
- UFS and JFS file system administration commands

<code>awk</code>	<code>bdiff</code>	<code>cat</code>	<code>chgrp</code>	<code>chmod</code>
<code>chown</code>	<code>cksum</code>	<code>cmp</code>	<code>comm</code>	<code>compress</code>
<code>cp</code>	<code>csch</code>	<code>csplit</code>	<code>cut</code>	<code>dd</code>
<code>df</code>	<code>du</code>	<code>expand</code>	<code>fbackup</code>	<code>frecover</code>
<code>file</code>	<code>find</code>	<code>fold</code>	<code>getaccess</code>	<code>grep/egrep/fgrep</code>
<code>head</code>	<code>hyphen</code>	<code>join</code>	<code>ksh</code>	<code>ln</code>
<code>ls</code>	<code>mv</code>	<code>od/xd</code>	<code>paste</code>	<code>prealloc</code>
<code>rm/rmdir</code>	<code>sed</code>	<code>sh</code>	<code>sort</code>	<code>split</code>
<code>strings</code>	<code>tail</code>	<code>tee</code>	<code>touch</code>	<code>uncompress</code>
<code>unexpand</code>	<code>uniq</code>	<code>uudecode</code>	<code>uuencode</code>	<code>wc</code>
<code>zcat</code>				

Complete details of the data types, structures, interfaces, and commands can be found in the *Large Files White Paper*.

## Behavior of Existing Applications

The current small file (less than 2GB) environment has not changed. The implementation has only introduced new interfaces that enable applications to manipulate large files. Therefore, as long as an application has no need to create, open, or `stat` large files, no code modification or recompilation is required. Also, because the existing environment is not changing, existing code that is recompiled on a system that supports large files will still behave as it did before the recompilation.

When an existing application that does not utilize the new interfaces to access large files encounters a large file, an error will be returned. These applications can expect both the `open()` and `stat()` calls to fail with `E_OVERFLOW`. This protection mechanism prevents applications unaware of large files from accessing a large file because the application may not be able to correctly handle large files.

There is also a file system-based protection mechanism to protect unaware applications from encountering large files. A file system can be designated as one of the following:

- **largefiles**—large files can be created on the file system
- **nolargefiles**—applications cannot create large files on the file system

The mechanism is administered through the file system administration commands:

Command	Description
=====	=====
<b>newfs(1M)</b>	Creates file systems and now accepts a flag to designate the file system either "largefiles" or "nolargefiles" (the default is "nolargefiles").
<b>mount(1M)</b>	Also accepts the "largefiles" or "nolargefiles" options.
<b>fsadm(1M)</b>	Modifies this attribute of the file system after it has been initially designated.

Full details can be found in both the white paper and associated manpages.

## Programming for Large Files

Large files are supported in two programming/compile environments. In the first, existing data types keep their old size and large files are accessed by explicitly calling the new 64-bit routines and using the 64-bit data structures and types. These new interfaces enable the full access of files up to the supported limit. Current 32-bit interfaces keep their old sizes and continue to work on files less than 2GB. The new and old interfaces can be mixed in the same program. Specific compilation flags are required to make these 64-bit interfaces available:

Compilation Flag	Provides
=====	=====
<b>_LARGEFILE_SOURCE</b>	<b>fseeko() and ftello() interfaces</b>
<b>_LARGEFILE64_SOURCE</b>	<b>all *64() interfaces</b>

In the second environment, existing standard data types and routines use 64-bit fields, allowing programs using standard POSIX and XPG interfaces

## 2-22 Major Changes for HP-UX 10.20



to support large files. Again, compilation flags are used to determine the interfaces sizes:

Compilation Flag	Provides
=====	=====
_FILE_OFFSET_BITS=32	32-bit environment (the default)
_FILE_OFFSET_BITS=64	64-bit environment; causes 32-bit routines and data structures to be 64 bits

Complete details of the programming/compile environments can be found in the white paper.

## Using Large Files

As stated above, if your application does not need to operate on file data beyond 2GB, you do not have to make changes to the application. The built-in protection mechanisms will help prevent the unaware application from accessing a large file. However, if the application needs to support the large file interfaces, there are a few alternatives to consider:

1. Make the program aware of large files without actually supporting them. Look for the new `open()` and `stat()` errors and, if encountered, take appropriate actions. For example, if your application is a browser that currently exits when `stat` fails, it can instead be changed to display an appropriate message and then continue.
2. Use standard types and calls, but compile the program in the environment that uses 64 bits for the relevant types. Note that other source changes will most likely be necessary to correctly support the larger data types.
3. Use the new `*64()` calls in the 32-bit compile environment. This involves explicitly coding the `*64()` names into the source code. This should only be done when a specific portion of the application needs to be large file aware, while other parts do not. This approach is not recommended if the alternative described above can be used.

Complete information can be found in the white paper.

Regardless of which interfaces are used (either the standard POSIX interfaces or `*64()` interfaces), all exhibit the same behavior on files less than 2GB. For example, the `*64()` interfaces can be used on small files just as they are used

on large files. A small file that was created/modified by a large file application can be modified/accessed by a small file application, and vice versa. There are no incompatibilities between the two types of interfaces when operating on files less than 2GB.

### **Eoverflow Error**

With large files, it is possible for a developer to mix and match 32- and 64-bit interfaces and data types. For this reason, many of the routines can return a new error, **Eoverflow**, indicating that the operation or returned data is too large for the environment or passed data structure. This can occur when a 64-bit call is made using a 32-bit data structure, or when an unconverted application tries to open a large file. The details of the error associated with each interface can be found in the interfaces' manpage.

### **Commands Support**

The commands listed in the above "Commands" subsection support operations on large files. In addition, commands that display or accept as arguments byte, word, and line counts also handle values larger than 2GB. For example, **wc(1)** displays byte, word, and line counts that can be larger than 2GB; **find(1)** accepts size arguments that are larger than 2GB, and **compress(1)** operates on files larger than 2GB. An exception to this general rule is **tail(1)**, which is (and has always been) limited to a 20K buffer for display. Therefore, **tail(1)** only displays up to 20K bytes of data, regardless of the line number argument that is passed. See the manpage for complete details.

### **Shells Support**

All three shells (**sh**, **ksh**, and **csh**) support redirection of data to/from large files.

### **Backing Up Large Files**

The **dd**, **fbackup/frecover**, and **vxdump/vxrestore** commands support large file archival and retrieval. The **fbackup** and **vxdump** data formats have been changed to accommodate the larger file sizes. The archives created by these versions of the commands are readable only by the corresponding **frecover** and **vxdump**, and not by their earlier versions. Archives created by older

## **2-24 Major Changes for HP-UX 10.20**

versions of `fbackup` and `vxdump` are readable by the new versions of `frecover` and `vxdump`.

Standards-conforming backup commands such as `tar`, `cpio`, `ftio`, and `pax` do not support archiving files greater than 2GB. These commands have limitations associated with standards and interoperability that prevents their support of large files. The manpages contain complete details.

The `dump/restore` commands do not support files greater than 2GB.

## Performance

There can be some performance degradation due to the use of 64-bit data types, which, on 32-bit architectures, require more instructions to manipulate than 32-bit types.

## Compatibility

The solutions for large files introduces new functionality. All prior functionality is preserved. The implementation is fully binary compatible with prior releases of HP-UX.

---

## Large User IDs

For 10.20, HP-UX will support user and group IDs ranging from 0 (root) to a maximum of `MAXINT - 1`, which is equal to 2,147,483,646 or  $(2^{31}) - 2$ .

The constant `MAXUID`, which defines the lowest integer that cannot be used as a user or group ID number, is raised from its current value of 60,000 to `MAXINT`. These constants have also changed:

Constant	New Value
<code>ACLUNUSED</code>	-35
<code>ACL_ANYGROUP</code>	-25
<code>ACL_ANYUSER</code>	-25
<code>ACL_FILEGROUP</code>	-26

```

ACL_FILEOWNER          -26
ACL_MSGROUP            -36
ACL_NSUSER             -36
UID_EUID               -34
UID_NOBODY             -2
UID_RUID               -33
UID_SUID               -32
=====

```

The formats of the HFS and VxFS file systems will change to allow storage of these new, higher user and group ID numbers.

The output format of the `audisp(1M)`, `ipcs(1)`, `ps(1)`, and `top(1)` commands will also change to make room for longer user and group ID numbers.

## Impact

A file system conversion is needed for HFS and VxFS file systems to support large user and group IDs (that is, those that are above the old maximum of 59,999). For HFS file systems, this conversion happens automatically when the first reference to a large user or group ID is created on that file system. For VxFS file systems, this conversion happens as part of the process of upgrading to VxFS version 3 (using the `vxupgrade(1M)` command). HFS and VxFS conversion to support large user and group IDs is not reversible.

The following table summarizes the file system interoperability for large UIDs.

```

=====
File System Type      | Will it work on a | Will it work on a
                      | pre-10.20 system? | Large UID system?
=====
HFS with no large UIDs* |      yes          |      yes
-----
HFS with large UIDs    |      no**         |      yes
-----
VxFS version 2         |      yes          |      yes***
-----
VxFS version 3         |      no           |      yes
-----
Third-party file system |      yes          | changes by vendor
                      |                  | are required
=====

```

\* This is only for HFS file systems that never had large

```
UIDs. HFS file systems that have at any time contained
large UIDs are assumed to still contain large UIDs.
** Mounting will be prevented by software only on HP-UX 8.x and
later.
*** Large UIDs cannot own files.
```

Third-party file systems will be broken by the large UID changes to the `vnode` layer.

Scripts relying on the exact output format of the `audisp(1M)`, `ipcs(1)`, `ps(1)`, or `top(1)` commands might not work. Scripts relying on exact field widths of the output of such commands are not, in general, supported.

Both the accounting file formats and the accounting commands are changing. This implies that the accounting commands cannot be used to examine accounting files generated on an earlier release of HP-UX.

Most applications will work as expected. However, there are some applications that might work incorrectly, unpredictably, or in a way that might introduce breaks in security. Only applications with the following might work unexpectedly:

- The system has a large user or large group ID defined. (Systems that do not make use of a large user or group ID will not be affected.)
- The application is compiled on a version of HP-UX that does not support large user IDs.
- The application is made up of one or more of the following:
  - Storage of user or group IDs in 16-bit (short) variables
  - Output of a user or group ID into a field too narrow to hold the new maximum
  - The `getaccess(2)` system call is used (this is an HP-proprietary extension)
  - File access control lists (ACLs) are used (an HP-proprietary extension and the most likely source of breaks in security).

Again, all other applications should work as expected.

Mixed-mode NFS environments (that is, environments that contain some large UID-capable machines and some pre-large UID machines) will produce unexpected behavior, as described in the table below:

	Will it work on a Small UID Client? (HP or third-party)	Will it work on a Large UID Client? (HP or third-party)
Small UID Server running HP-UX	yes	Patches needed*
Small UID Server (third-party)	yes	Determined by server vendor
Large UID Server (HP or third-party)	Patches needed**	yes

\* Patches are needed because large UIDs cannot own files on the server. Without patches to the server, large UID clients might introduce breaks in security in the server.

\*\* The client sees the wrong owners for large UID-owned files. Without necessary patches to the client, small UID clients will introduce breaks in security to large UID servers.

The patches needed for both the server and client are:

- Release 9.0—PHKL\_6758
- Release 9.01—PHKL\_6759
- Releases 9.03, 9.05, and 9.07—PHKL\_6760
- Release 9.04—PHKL\_6761
- Release 10.0—PHKL\_6762
- Release 10.01—to determine availability, see the website <http://us.external.hp.com/> and click on “Browse Patches”.
- Release 10.10—to determine availability, see the website <http://us.external.hp.com/> and click on “Browse Patches”.

The changes to `libc` needed for large user ID support will be made using intra-library versioning (not SysV versioning). So, applications compiled on a large user ID system will see the new behavior, while old applications will not. All HP-supplied libraries that are impacted by this versioning have themselves been versioned appropriately. However, third-party library suppliers might need to version their libraries to support both large and small user ID environments. Such versioning is needed if the library makes use of any of the constants listed above or any of these structure types:

## 2-28 Major Changes for HP-UX 10.20

- struct acct
- struct acl\_entry\_internal
- struct audit\_str\_data
- struct audit\_str\_hdr
- struct inode
- struct self\_audit\_rec

The `getpwent(3C)` functions will no longer translate out-of-range UID and GID values to `(UID_MAX+1)`; the values will be returned as is.

## Limitations

Although the full positive integer range can now be used for user IDs, system performance will suffer if large numbers of users or groups are defined on a single system.

HFS and VxFS file systems with quota checking enabled will maintain quota information only for users with user IDs less than 67,000,000. The `edquota(1M)` command will not permit quotas to be established for other users. This limitation will be removed in a future release.

Due to standards limitations, the `cpio(1)`, `ftio(1)`, `pax(1)`, and `tar(1)` commands cannot support the full range of UID/GIDs. These commands will maintain the old UID/GID limits of 60,000. Any files that have more than 60K UID/GIDs will be archived with `UID_NO_CHANGE/GUID_NO_CHANGE` and will therefore be restored under the UID/GID of the restoring process.

---

<b>Note</b>	<code>tar</code> searches for the archived user name in the <code>/etc/passwd</code> file and the group name in the <code>/etc/group</code> file and attempts to get the UID/GID from these files. If these are not found, <code>tar</code> attempts to establish ownership with the archived UID/GID. So, most files should be restored with the correct ownerships.
-------------	---

---

The `fbackup(1M)` tape format has been changed to accommodate the large UID/GIDs. The tapes created by this version of `fbackup` will be readable only by the matching version of `frecover(1M)` and not by older versions.

The `dump(1M)` tape format has also changed. If the file system being dumped has not contained large user or group IDs, `dump` will use the old tape format and will function as before. The archive created will be readable by older

versions of `restore(1M)`. However, when dumping a file system that has contained large user or group IDs, the new dump tape format will be used. The tapes created by this version of `dump` is readable only by the matching version of `restore` and not by older versions.

## Performance

Performance is not degraded on identical system when large UID functionality is added. Some commands have performance that varies with the number of users and/or groups defined on a system; these commands can expect to have degraded performance on systems where many users or groups are defined.

The performance of the disk quota subsystem is dependent on the highest user ID value to which a quota has been assigned. Systems with quotas defined for large UIDs will see performance degradation.

## Alternatives/Compatibility

Applications that do not work as expected, or applications expected not to work (as described above), can be fixed using the following steps:

1. Change any 16-bit (`short`) variables in which user or group IDs are stored into type `uid_T` or `gid_t` as appropriate. These types are defined in `<sys/types.h>`, `<sys/unistd.h>`, and so on).
2. Widen the output fields for user or group IDs to hold at least ten digits.
3. Recompile the application on a system that supports large user IDs.

Third-party file systems will need to recompile with the new kernel header files to adapt to the `vnode` layer changes.

Third-party libraries that need to be versioned should make use of the intra-library versioning facility to provide their existing behavior to old (pre-large-UIDs) applications and current behavior to current applications.



## **Obsolescence**

New HFS file systems will be created in the small user ID format and will be automatically converted to large user ID format on demand as described above. By default, new VxFS file systems will be VxFS version 3, so they will support large user IDs. The future direction is the large user ID format; support for mounting or using file systems in the small user ID format will be removed in a future release.

---

## **Multi-Processor Control**

For 10.20, `mpctl` determines how many processors are installed in the system and assigns processes/lightweight processes to run on specific processors.

This call can be used to increase performance in certain applications, but should not be used to ensure correctness of an application.

Much of the functionality of `mpctl` is dependent on the underlying hardware. An application that uses this system call should not be expected to be portable across architectures or implementations.

---

## **QIC Media**

As of 10.20, you cannot receive software updates on QIC. In addition, at 10.20, QIC is not supported as a backup device. Therefore, you should migrate to CD-ROM media for software installation and updating.

---

## Reverse ARP (Address Resolution Protocol)

For 10.20, a system can determine its IP address(es) at bootup time rather than relying on a hardwired configuration file.

Applications have been added to find IP addresses using the Reverse Address Resolution Protocol (RARP, RFC 903). A startup script has been modified to use this information when configuring network interfaces.

### Impact

Only users that turn on RARP functionality will see an impact. There are no performance or functionality issues.

### Compatibility

These RARP applications conform to the RARP protocol as specified in RFC 903. They are designed to be used interchangeably with RARP applications in other vendors' systems.

---

## SD-UX

Features:

- **Eliminates the need for the `-r` option for all commands—**

The `-r` command line option is no longer needed to specify that the SD operation is directed to a root directory other than `/`. The command will interpret the target specified to determine what type of root (primary or alternate) it is.

Backward compatibility is maintained by silently accepting the `-r` option if it is given. The determination of the target default value is unchanged if a target is not specified on the command line.

- **Eliminates the need for the `-l` option for `swremove` and `swconfig` commands—**

The `-l` command line option is no longer needed to specify that the `swremove` and `swconfig` operation is to operate on link-installed software. The command will determine whether software is link-installed or physically installed and behave accordingly. For backward compatibility, if the `-l` option is given, the session will only operate on link-installed software.

The `swinstall` command still requires the `-l` option to direct a linkinstall.

■ **The kernel is rebuilt and the system is rebooted when removing a kernel fileset—**

Removing a kernel fileset will now, by default, cause a kernel rebuild and a system reboot. The `autoreboot` option now applies to the `swremove` command and must be set “true” to effect a kernel build.

Compatibility with previous behavior is maintained by specifying the new `swremove` option `auto_kernel_build=false`. The option should be used only when removing a kernel fileset to delete files and not to remove functionality.

The behavior when removing kernel software from a NFS client system remains unchanged.

■ **Add cdrom as a unique source type along with the existing directory and tape types—**

This provides an associated default source device when you select the `cdrom` source type.

■ **Partial Internationalization (I18N) and Localization (L10N)—** You will see messages on the screen and in the SD log file displayed in the local language. This feature requires that the translated SD message catalog is available on the system and that the `LANG` environment variable in `etc/rc.config.d` has been changed to designate that local language. For example, to make SD agent and daemon log files display in Japanese, `etc/rc.config.d/LANG` must be `LANG=ja_JP.SJIS` or `LANG=ja_JP.eucJP`.

## Enhancements and Defect Fixes

The order in which products and filesets are loaded has been changed. The changed behavior causes the load order to conform to the original specification, which was not always the case in 10.10.

A Session file, which is written when an SD command is operating on a single local target, will not have the local target name included in the target specification.

Partial file recovery is enhanced to cope with sessions that compressed or uncompressed files.

## Impact

- The complexity of the command line is slightly reduced by eliminating redundancy.
- The complexity of the command line is reduced for the `swremove` and `swconfig` commands.
- The need to rebuild the kernel manually and reboot after removing kernel functionality from a standalone system or a server is eliminated.
- Use of CD-ROM as a source type is easier to specify.
- The capability exists to have users see the messages and log files in a language other than American English.
- Porting the Session file to be used by another system is easier because you no longer have to edit the target system specifier.
- The restriction no longer applies that partial file recovery can only apply to sessions that did not change the form of the files: compressing uncompressed files when copying a depot, uncompressing files when copying or installing from a depot containing compressed files.

## Performance

Performance of the command remains unchanged. The remove process will be affected by the following kernel build and reboot.

Magnetic tapes that were written using pre-10.20 commands might take longer to load. This is due to the 10.20 software expecting products to be in a sequence other than what is on the media.

## Compatibility

Existing scripts are not impacted. Making the deletions in the kernel when removing functionality involves supplying a “preremove” control script with the kernel fileset.

Some command syntax that used to fail will now work.

---

## System Compatibility

This section points you to tools and other sources of information you will need to upgrade from HP-UX 9.x to 10.x.

If you are upgrading from an HP-UX 9.x release to 10.01, 10.10, or 10.20, refer to *Upgrading from HP-UX 9.x to 10.x*, which details the steps you must take to upgrade. Also see this chapter, “Major Changes for HP-UX 10.20”, Chapter 3, “Major Changes for HP-UX 10.10”, and Chapter 4, “Major Changes for HP-UX 10.0 and 10.01” in this document for information on system changes that may require you to make changes to make your 10.x system work correctly.

If you are *installing* (not upgrading) 10.01, 10.10, or 10.20, upgrading from 10.0 to 10.01, 10.10, or 10.20, upgrading from 10.01 to 10.10 or 10.20, or upgrading from 10.10 to 10.20, see *Installing HP-UX 10.20 and Updating HP-UX 10.x to 10.20*.

---

## VM Kernel Address Space

HP-UX 10.20 contains a fundamental change in the virtual memory kernel address space that might affect third-party kernel components, such as kernel drivers or file systems. Third-party HP-UX kernel components might contain code that makes assumptions that might no longer be true, resulting in a need to modify these kernel products in minor ways. This change was done to enable other system performance optimizations. Only kernel code and kernel component products are affected. If your product contains only user level code, like typical ISUs/ISVs, you are not affected by this change.

More specifically for kernel component developers, the space id associated with the kernel buffer cache will be changing from 0 (`KERNELSPACE`) to a non-zero value. The buffer cache will still remain in the third quadrant, but the value of `sr6` will no longer be zero. This means that short pointer accesses within the kernel address space will not be affected. However, routines that use long pointers (that is, where a space is specified) will need to make sure that they are not currently hardwiring the space to zero (or `KERNELSPACE`).

Previous to this change, the space id associated with both the first quadrant and third quadrant for the kernel address space was zero (the value of the constant `KERNELSPACE`). After the change, the space for the buffer cache is non-zero and it is contained in the global variable `bufcache_spaceid`.

Until now, the design assumption has been that both quadrants are in the same space and code that did not use the entire long address could hardwire `KERNELSPACE` for the space identifier.

Except for remote file systems like NFS, this has no affect on networking drivers. Mbufs are allocated out of dynamic memory in the first quadrant, whose space will remain zero. It is only when a data pointer might point to an address in the buffer cache that there might be a problem. This can happen in the networking write path when, for instance, NFS is providing file data to a remote system.

The write path through the network drivers needs to be checked for calls to routines that take a space identifier as an argument. Some examples of routines like this are: `lbcopy`, `privlbcopy`, `dma_sync`, `fdcache`, `wsio_map`, `sio_map`, `ltor`, and so on. Typically there can be multiple calls in the driver to some of these routines, but only one or two need to be changed because they are the

only ones where data from outside of the networking subsystem is passed to the network interface.

In some cases, the driver already has a variable for the space argument, but that variable has been hardwired to `KERNELSPACE`. For example, the only change to the `lan1` driver is follows:

**Old:**

```
sid = KERNELSPACE;
mbuf_vaddr = mtod (mbuf_ptr, char *);
```

**New:**

```
mbuf_vaddr = mtod (mbuf_ptr, char*);
sid = ldsid(mbuf_vaddr);
```

This change is because `sid` was used as an argument for `ltor()`, `privlbcopy()`, and `fdcache()`, so it only had to be assigned the correct value. The `ldsid()` procedure is an assembly language procedure that simply uses the `ldsid` PARISC instruction to obtain the proper space associated with the virtual address passed in as an argument.

Other drivers might need more changes if a variable was not used already as an argument to routines like `privlbcopy`; that is, `KERNELSPACE` was passed in as an argument. In this case, a variable like `sid` in the above example should be declared and `ldsid()` should be called to initialize it. The argument to `ldsid` should be the virtual address associated with the data being written to the network interface. Then, all the hardwired `KERNELSPACES` associated with this virtual address should be replaced with the newly declared variable.

Once the driver has been fixed, it should be tested on a 10.20 kernel by attempting to serve an NFS client over the network (that is, another machine should attempt to mount one of local file systems via NFS). If the appropriate changes have not been made, the system will immediately crash. If a debug kernel is used, you might get an assertion failure. Otherwise, you will most likely get a data page fault, where the `ior` is in the third quadrant (`0x80000000` through `0xBFFFFFFF`), but the `isr` is 0.

---

## VME Services

The VME Services product provides configuration tools, a kernel-level driver, and user- and kernel-level VME bus access functions to install, administer, and access VME bus functionality for application-level programs and application-level and kernel-level drivers. Generally, the `vme2` kernel driver supports programming of the VME bus adapter to the VME bus on behalf of the caller, arranging for bus transfers; mapping memory between the HP-UX host and VME bus address space (and vice versa); interrupt handling; reading and writing of EEPROM VME data; and so on.

For 10.20, VME Services has been updated to provide additional function calls for asynchronous DMA purposes relating to queues, status, and interrupts; for mapping pages and buffers to the VME bus, and remapping memory; and for protecting access to the data and functions involved.

These new functions are side effects of increasing VME functionality to include a generally invisible enhancement to support HP-UX-to-HP-UX backplane communications via a new socket IPC domain `AF_VME_LINK`.

## Performance

Performance impact is isolated to hardware systems that include VME support. The performance of VME systems at 10.20 will be equal to or better than the performance at 10.10 for similar functions.

## Supportability

Current firmware booting and CPU-number-to-LAN-address-mapping are supported without change. UDP `bootp` requests over the backplane are also supported.

Only some socket calls are supported at 10.20. See the *HP-UX 10.20 VME Services Guide* manual for details.



### **Installability**

VME Services are automatically installed onto a system, but can be manually added to the kernel as in previous releases.

### **Binary Compatibility**

VME drivers that work at HP-UX 10.10 will not need to be recompiled or rebuilt.

Configuration files will have to be revised for drivers or other applications that use pre-10.20 BPN.

### **Standards**

VME Services implements latest approved IEEE 1015-1987 standard as supported by the different hardware levels on which VME Services runs.



## Major Changes for HP-UX 10.10

---

This chapter describes the following HP-UX 10.10 changes:

- System Compatibility
- System Configuration
- Large File System
- Large Physical Memory
- Large Process Data Space
- Native Language Support (NLS)
- 3D Graphics
- Upgrading from 9.x
- PCx-Processor Support Pre-Enable
- Distributed Interrupts and I/O Forwarding
- Logical Volume Manager (LVM)
- Shared Logical Volume Manager (SLVM)
- UNIX95
- HP Common Desktop Runtime Environment (CDE 1.0)
- NIKE Array Enhancements
- 60K File Descriptors
- Dynamic Host Configuration Protocol (DHCP)
- Additional Tools
- VME Services
- Large Buffer Cache
- pstat(2)
- Remote Process Call (RPC) Interfaces
- TCP/IP Transport
- Cold Install
- Software Distributor (SD)
- Known Problems

Also see the following for additional 10.10 changes:

- **Chapter 5**, “Other Operating System and Subsystem Changes”:
  - Distributed Computing Environment (DCE)
  - File System Changes
  - Graphics
  - HP VUE
  - Kernel Configuration
  - Kernel Tunables
  - Multimedia
  - Peripheral Devices
  - Printing with SharedPrint/UX
  - Programming Languages
  - SAM
  - UUCP
  - X Windows
  - HP VUE
- **Chapter 6**, “Networking”:
  - 48 LAN card support
  - Sockets
  - Streams
  - XTI
- **Chapter 7**, “Commands and Libraries”:
  - Changed Commands in Section 1
  - Changed Commands in Section 1M
  - Library Routines (libc)
  - Changes to Other Commands and Files

---

## System Compatibility

### Upgrading from HP-UX 9.x to 10.x

This section points you to tools and other sources of information you will need to upgrade from HP-UX 9.x to 10.x.

If you are upgrading from an HP-UX 9.x release to 10.01 or 10.10, refer to *Upgrading from HP-UX 9.x to 10.x*, which details the steps you must take to upgrade. Also see this chapter, “Major Changes for HP-UX 10.10” and Chapter 4, “Major Changes for HP-UX 10.0 and 10.01” in this document for information on system changes that may require you to make changes to make your 10.x system work correctly.

If you are *installing* (not upgrading) 10.01 or 10.10, upgrading from 10.0 to 10.01 or 10.10, or upgrading from 10.01 to 10.10, see *Installing HP-UX 10.10 and Updating HP-UX 10.01 to 10.10*.

Upgrading from 9.x to 10.10 is a two-step process:

1. Upgrade to 10.01 (also load 10.01 applications and any optional 10.01 features you need)
2. Upgrade to 10.10

*Upgrading from HP-UX 9.x to 10.x* details the above steps, as well as necessary preparation steps.

Some 9.x systems cannot be upgraded to 10.01 with the upgrade tools supplied with the initial 10.01 release. These systems are:

- HP-UX “DUX” clusters
- Systems running HP-UX 9.07
- Series 700 Model J systems
- C100/110

These systems can now be upgraded using the tools and documentation in the package “HP-UX Upgrade Tools for 9.x to 10.x”. However, if you are running on a 9.07 system, you must upgrade all the way to 10.10; you cannot stop at 10.01. All of the other systems listed above can be upgraded to 10.01 or 10.10 (via 10.01). Again, see *Upgrading from HP-UX 9.x to 10.x* for details.

### **Compatibility with Systems Before HP-UX 10.10**

Applications you develop taking advantage of new 10.10 features will not necessarily run on earlier systems.

---

## System Configuration

### New System Configurations (S700/800):

- T520
  - 120MHz
  - 1-12 CPUs
  - main memory: 256MB - 3.75GB
  - uname: 9000/892
- C110
  - 120MHz
  - 32-512MB

### Pre-enabled Hardware

- Decoupled hardware—the SPU requires no system software changes.  
Systems:
  1. S800 PCx-T' and PCx-L based systems
  2. S700 PCx-T' based systems
- Prototype coupled—the SPU requires moderate risk changes to the system software. Systems:
  1. PCx-U S800 servers
  2. PCx-U S800 servers
  3. PCx-U S700 systems
  4. PCx-L2 and PCx-U S700 systems

### Supported NFS Diskless Configurations

- Servers:
  - 705, 710, 715, 720, 725, 730, 735, 745, 747, 750, 755, J-series, PCx-T', 8\*7, 890 1-4 way, D-series, E\*5, F\*0, G\*0, H\*0, I\*0, K-series, T500 1-12 way, T520 1-12 way, PCx-U servers
- Clients:
  - 705, 710, 715, 720, 725, 730, 735, 742, 743, 745, 747, 750, 755, J-series, PCx-T'

---

**Note** S800 CIO and S8\*2 servers are not supported.

---

---

**Note** You cannot install both a 10.10 and a 10.01 system simultaneously on an NFS diskless cluster.

---

### **Product Stability Notice**

The HP-UX 10.10 Release is the last version of HP-UX which will support the following systems which have been previously discontinued from the HP 9000 product line. All systems listed here will continue to be supported on HP-UX 10.10 until the completion of their guaranteed support life. The guaranteed support life for a system is 5 years following the removal (discontinuance) of the product from the Corporate Price List.

Users preferring to take advantage of future HP-UX operating system release functionality should now actively work with their account representatives to discuss the upgrade paths from each of these systems to the latest members of the HP 9000 product line. These upgrade paths include trade-in credits for existing systems and the ability to transfer software licenses to the new platform.

The systems that have reached Operating System Functional Stability with this release include:

840S, 825S, 835S/SE, 845S/SE, 850S, 855S, 860S, 865S, 870S/x00, 822S, 832S, 842S, 852S, 890

With this change, HP-UX 10.10 will also be the last release to support Battery Backup and CIO functionality. The Battery Backup capability has been surpassed in functionality by developments in Uninterrupted Power Supply technology. Please see your account representative for information on UPS solutions available from Hewlett-Packard.

### **3-6 Major Changes for HP-UX 10.10**



---

## Large File Systems

For 10.10, the maximum size of HFS and JFS file systems has been increased from 4 Gb to 128 Gb. (The maximum size of files continues to be 2 Gb.)

Large file systems are supported by HP-UX, including file system administration commands (HFS and JFS), SAM, SD, LVM, accounting commands, quota commands, and backup commands. These interfaces, as in previous releases, allow you to create, modify, and recover file systems. All of the command interfaces remain the same; there are no new options for large file systems. The only difference is the parameter values may now be larger to accommodate file systems greater than 4 Gb.

Large file systems can be exported through NFS to other clients, who are be on any other supported version of HP-UX. NFS clients do not have to be on 10.10 to read/write an exported file system. For example, a 10.10 server can export a 20 Gb file system to 9.x clients through NFS and each of the clients can read/write to the 20 Gb file system. Non-HP NFS clients can also access an LFS on an HP server.

## Making Small File Systems Large

Increasing the size of a small file system to a large file system is supported through the standard file system extension command `extendfs(1M)`. Using `extendfs` for JFS is new for HP-UX 10.10. An alternative to `extendfs` is to back up the data, remake the file system, and restore the data.

## Compatibility with pre-10.10 Filesystems

All file systems built with earlier versions of HP-UX are fully supported in 10.10. There is no conversion required of pre-10.10 file systems to be functional in 10.10.

## Limitations

It is not possible to use large file systems for root or boot disks because there are limitations of 2 Gb or 4 Gb (depending on the processor) in the IODC.

Applications that access devices directly will continue to have a 4 Gb limit. There is no interface for large devices. The implementation is for large file

systems only and is supported through the standard set of administration commands mentioned above.

It is possible for users to access file systems directly through their respective device files. The contents can be copied with `cp(1)` or displayed with `od(1)`. This usage is rare and will continue to be limited to 4 Gb. An exception to this is `dd(1)`, which will support large file devices.

## **Performance**

For 10.10, there should be no performance degradation. Building and recovering (using `fsck`) a very large HFS file system will take proportionately longer than a smaller system. For a JFS file system, the scaling is proportional to the log size, not the file system size. Random access performance on both file systems is not impacted.

## **dcopy Command**

The `dcopy` command has not been enhanced to support large file systems, but will continue to operate on small file systems. All other commands and interfaces remain intact.

## **fsirand**

`fsirand` supports the `lseek()` block modifications for large file systems. If `fsirand` detects a large file system, all `lseek()` calls must use block or byte offsets.

Previous versions of `fsirand` will not work properly on a large file system. The current 10.10 version of `fsirand` should not be run on a previous release.

---

## Large Physical Memory (3.75 Gb)

HP-UX 10.10 now supports up to 3.75 GigaBytes (Gbs) of physical memory on the Series 890, T500, and T520. (Physical memory sizes up to 2 Gbs are supported on S700 platforms.) This fully utilizes the PA-RISC 1.1 32-bit architectural limit of 3.75 Gbs of physical memory. The remaining .25 Gbs is reserved for I/O devices.

Prior to this release, only at most 2 Gbs of physical memory could be installed using eight 256 MB memory boards. With the introduction of a new high-density .75 Gb memory board, a total of 3.75 Gbs can be installed using only five boards.

Note that application performance may not necessarily improve with the increased physical memory. See the Performance section below for details.

What changed:

1. Changed necessary core kernel areas to give user applications access to the entire 3.75 Gbs. Only the physical addressing range has been extended; there is no increase to any of the existing HP-UX 10.0 process virtual memory limits. For example, process text and data are still limited to 1 Gb segment sizes (although this can be increased to 1.9 Gb for combined text+data using the `EXEC_MAGIC` compiler option). The total amount of System V shared memory and shared Memory Mapped Files is still limited to 1.75 Gb. Only the file system buffer cache has been increased (see below).
2. Modified kernel dump and `savecore(1M)` compressed core file scheme for managing larger (up to 3.75 Gb) core dumps. See the section `savecore(1M)` in Chapter 6 for details.
3. Modified kernel level debuggers (such as `adb` and `q4`) for analyzing the compressed core files.
4. Increased file system buffer cache capacity from .9 Gbs to 3.75 Gbs. See the Performance section below for details.
5. Minor changes to the following kernel tunable parameters:

- `unlockable_mem`

The `unlockable_mem` tunable specifies the amount of memory that can never be locked down by user processes. This memory is effectively

reserved for virtual memory demand paging. A value of less than or equal to zero for this parameter indicates that you want the system to choose a suitable default.

`unlockable_mem` was previously specified as a byte quantity. To support greater than 2 Gbs of physical memory, the definition of `unlockable_mem` has been changed to specify the number of 4 Kb *pages* that cannot be locked down.

The SAM utility has been changed to reflect this.

#### ■ `eqmemsize`

The `eqmemsize` tunable defines the size, in pages, of the equivalently mapped memory reserve pool. An equivalently mapped page has the same address in both real and virtual processor modes. (Only pages contained in the first Gigabyte of the physical address space can potentially be used as equivalently mapped pages.) Equivalently mapped memory is used primarily by drivers for processing I/O requests, and, beginning with this release, for the support of address aliasing requests issued on behalf of `EXEC_MAGIC` processes. See the section on `EXEC_MAGIC` executables for details.

The system handles requests for equivalently mapped memory by dynamically obtaining a free page with a matching virtual address from the system wide memory pool. If the system cannot dynamically obtain an equivalently mapped page, it goes to its reserve pool. Normally this reserve pool should never be exhausted. However, those systems with a relatively high load and/or a physical memory configuration that exceeds 1 Gb may potentially deplete this reserve pool.

Given the increased demand for equivalently mapped memory and the fact that HP-UX now supports larger memory configurations, the default size of the reserve pool has been increased from 10 pages to 15 pages. In addition, the `eqmemsize` tunable parameter now specifies the *minimum* size of the reserve pool in pages. At boot time, the system will further increase the size of the reserve pool beyond its minimum based on the amount of physical memory installed.

The above changes to the equivalently mapped reserve pool scheme will help to increase the availability of equivalently mapped memory. However, depending on the exact nature of your application, system load, memory

### 3-10 Major Changes for HP-UX 10.10

and I/O configurations, and so on, the reserve pool may still become exhausted. If this happens, the system will print a message to the console indicating that the reserve pool has been exhausted and that `eqmemsize` should be increased.

## Impact

For large MP systems, the overall boot time should not increase significantly with additional physical memory. This is because the memory self-test portion of the boot sequence is executed in parallel across all available processors.

The time for the system to dump core after a panic/TOC will increase with the added physical memory. As a general rule, the core dumping times should scale linearly with increased physical memory. For example, the time to core dump a 3.75 Gb system should be approximately double the time needed to core dump a 2 Gb system (all things being equal).

The overall time for the `savecore(1M)` utility to subsequently run after a system panic/TOC will increase significantly if the core file compression option is utilized or if `savecore(1M)` chooses to perform compression. Compression can be explicitly disabled if sufficient disk space is available to contain the resulting core dump. See the section `savecore(1M)` in Chapter 6 for details.

## Performance

There is no performance degradation on existing applications for physical memory sizes less than or equal to 2 Gbs.

Measurable performance change beyond 2 Gbs of physical memory *might* occur for the following kinds of applications:

- Memory limited applications paging/swapping at 2 Gbs

Those applications that are currently memory-limited at the current 2 Gb maximum and experiencing excessive VM paging/swapping activity can expect a performance boost with the added physical memory. Examples of such applications include general multi-user systems such as large T500 X-terminal servers, and so on.

- Local file system intensive applications

Those applications that are heavily accessing large amounts of local file system data (that is, not remotely accessing files across the network) can expect to see a performance boost *if* they currently have a 2 Gb system and find they need a .9 Gb buffer cache for optimum performance. Because the file system buffer cache has now been extended from its current .9 Gb capacity limit to (at most) 3.75 Gb, any applications currently pushing the .9 Gb limit should see an improvement. Examples of such applications include running database packages (such as Oracle) over the file system, rather than over the raw device path.

Note that the added physical memory beyond 2 Gbs will be of little or no benefit to the following kinds of applications:

- OLTP database servers using the raw device path (not the local file system) for accesses to on-disk data. This is because the performance of the such database applications is limited by the amount of System V shared memory available (1.75 Gb).
- NFS file server applications (as measured by the LADDIS benchmark). This is because the file system buffer cache size for large systems is insignificant to the overall performance of LADDIS as the number of IOPs/sec becomes large. Hence, a larger buffer cache and increased memory beyond 2 Gbs will have very little (if any) performance advantage.

The overall run-time size of the HP-UX kernel will increase to accommodate larger memory sizes in excess of 2 Gbs.

---

## Large Process Data Space

For 10.10, the current `EXEC_MAGIC` implementation has been enhanced to provide the following features:

- text is now shared between multiple copies of the same `EXEC_MAGIC` executable.
- true copy-on-write semantics are now provided for `EXEC_MAGIC` text pages.
- lazy swap reservation is now provided on the `EXEC_MAGIC` text segment.
- `EXEC_MAGIC` text is now loaded in on demand instead of load-on-exec.
- null reference semantics are now provided for `EXEC_MAGIC` executables.

`EXEC_MAGIC` executables are able to access >.9Gb of process private data because data is allowed to immediately follow text in quad 1. By allowing text and data to occupy the same quadrant, they must have the same space. On the other hand, process private data must have a unique space id. Therefore, in the previous implementation of `EXEC_MAGIC`, text was actually viewed as a part of the data. Because HP-UX currently supports only one virtual translation (that is, <space, offset> pair) to each physical page, `EXEC_MAGIC` text could not be shared between multiple processes. The new implementation of `EXEC_MAGIC` provides read-only aliasing, which allows multiple addresses to map the same physical page.

Because only one process actually owns a translation to a page, true copy-on-write is not currently implemented on HP-UX. When a second process attempts to read or write a shared page, the second process will receive its own private copy of the page. With read-only aliasing, processes can share a text page with different virtual addresses if they are only reading the page. A process will receive its own private copy of the page only when a write is performed.

Because `EXEC_MAGIC` text segments were considered part of the data segment, the text segment was writable. Because HP-UX guarantees swap space will be available whenever a process requires, swap space was reserved for the entire text segment at `exec()` time. Because most users do not write to their text segment, the swap space reserved for the process is never used. To make more efficient use of swap space, a lazy swap reservation policy is implemented for `EXEC_MAGIC` text. Swap space is only reserved for those pages that are written to.

`EXEC_MAGIC` text is entirely loaded in at `exec()` time to protect against `a.out` modifications while the application is being run. The new implementation of `EXEC_MAGIC` will guard against this happening by marking the `EXEC_MAGIC` executable `VTEXT`. This allows the text to be demand loaded.

Previously null reference semantics were not supported on `EXEC_MAGIC` executables because text was assumed to start at address zero. This functionality is now provided.

## Impact

- You should see more memory and swap space available when running multiple copies of the same `EXEC_MAGIC` executable because unmodified text pages are now shared.
- You should see a faster startup time for `EXEC_MAGIC` executables because text is now demand paged instead of being entirely loaded at `exec()` time.
- You could see faster execution time on `EXEC_MAGIC` executables because pages are now only copied on write instead of being copied on any access.
- You could see `EXEC_MAGIC` application failure while running if swap space is not available when attempting to write to a page of text. If you do not use self-modifying code, you will not see this failure.

## Performance

Performance should be better than the previous implementation of `EXEC_MAGIC` due to the demand paging and copy-on-write features. Performance on `SHARED_MAGIC` executables should not be affected.

## Alternatives/Compatibility

Due to the new lazy swap reservation policy, `EXEC_MAGIC` executables can now fail in the middle of execution due to lack of swap space. One workaround is to add more swap space.



---

## Native Language Support (NLS)

The following commands and libraries have been internationalized in a codeset-independent manner and can now support up to 4-byte EUC, as well as all other HP-UX supported single-byte and multi-byte codesets.

```
=====
DESCRIPTION                                COMMAND NAMES
=====
```

File/Text Manipulation	bdiff, comm, compress, csplit, deroff, diff3, diffmk, dircmp, expand, fold, head, pr, split, tail, tee, uncompress, unexpand, wc, xargs, zcat
File System	chgrp, chmod, chown, df, file, fsck, fsdb, link, touch, unlink
File Management	fbackup, frecover
Communication/Network	uupick, uuto, uux, write
Program Development	asa, tsort
Source Control	admin, delta, get, prs, rmchg, rmdel, sact, sccs, unget, val, what
System Administration repquota	getconf, quota, quotacheck, quotaoff, quotaon,
Message Handling	findmsg, findstr, insertmsg
Shells	csh
Miscellaneous	at, basename, batch, calendar, crontab, date, env, getopt, mesg, nice, nohup, pathchk, pwd, tabs, tput, which, who
Libraries	xcurses (UNIX95)

```
=====
```

## **Impact**

The commands listed above are now fully internationalized and will now support all HP-UX single-byte and multi-byte codesets.

## **Performance**

There is no significant performance degradation.

## **Compatibility**

HP is committed to investment protection through compatibility and attempts to eliminate all incompatibilities. Any compatibility issues with any of the above commands are discussed under the individual command/interface sections.

## **Alternatives**

4-byte EUC will be enabled with the above commands in a codeset-independent manner that will continue to support other HP-UX supported codesets. You can select the language/codeset by setting the appropriate NLS environment variables.

## **Localization**

The message catalogs for the following commands

- `bdiff`
- `compress`
- `csplit`
- `diff3`
- `file`
- `head`
- `mesg`
- `nice`
- `tail`
- `touch`
- `tsort`
- `xargs`

will be newly localized into Japanese, Traditional Chinese, Simplified Chinese, Korean, German, French, Italian, and Spanish.

## Future MSE Changes

ISO/MSE 9899:1990/Amendment 1:1994(E) defines extensions to ISO/MSE:1990 Programming Language - C. The amendment provides a more complete and consistent set of utilities for application programming using multi-byte and wide-character functions, as well as providing alternate spellings for language-specific tokens. ANSI C has since adopted this amendment and X/Open will eventually align with this amendment. HP will comply with the MSE amendment in a future release.

## Compatibility

There are a number of differences between the MSE and HP-UX 10.0 XPG4 interfaces. The following table lists the differences.

HP-UX 10.0 XPG4	MSE
(f)printf()/(f)scanf() use %C, %S to specify wide string, wide char	f()/(f)scanf() use %ls, %lc to specify wide string, wide char
(f)putwc(), putwchar() first argument is wint_t	(f)putwc(), putwchar() first argument is wchar_t
wcschr(), wcsrchr() second argument is wint_t	wcschr(), wcsrchr() second argument is wchar_t
wcstok() only two arguments are specified	wcstok() third argument is wchar_t **
wcsftime() third argument is char *	wcsftime() third argument is wchar_t *

HP-UX has a strong commitment towards investment protection and will make every effort to support source and binary compatibility for applications. HP is tracking these standards closely and will provide a migration plan consistent with our commitment to investment protection and open standards.

## Locale cs\_CZ.iso88592 Change

The locale `cs_CZ.iso88592` has changed to correct the `LC_TIME abday` abbreviation for Thursday. The change is from `C(acute)t` to `C(caron)t`. Because the abbreviated name for Thursday is now different, applications and commands that retrieve the `ABDAY_5` from the locale will now display a different value.

An alternative to retrieving different values is to put the `abday` defect back on your system. See the `localedef(1M)` manpage for details.

---

## 3D Graphics

Features:

- Support for HP VISUALIZE graphics and E&S Freedom Series graphics.
- Starbase has changed to include the new texture mapping feature, and no changes have been made to the other APIs.
- Support for remote HP PEX.
- Support for 3D graphics using the HP VMX driver (for the S800).

As in previous releases, Starbase does not support any bitmapped devices on the S800. For 10.10, applications can now run remote graphics from a S800 machine. Specifically supported is the 3D VMX driver and PEX protocol sent to remote S700 machines.

---

## Upgrading from 9.x

The 9.U3 upgrade tool, an enhancement of the existing HP-UX 9.U2 upgrade tool, allows you to easily and efficiently upgrade your HP-UX 9.x-based systems to HP-UX 10.x.

9.U3 supports the following upgrades:

- all upgrades supported by 9.U2
- 9.05/9.07
- DUX clusters

### 3-18 Major Changes for HP-UX 10.10

- HP VISUALIZE Graphics

See *Upgrading from HP-UX 9.x to 10.x* for detailed information on upgrading from an HP-UX 9.x release to 10.01 or 10.10.

If you are *installing* (not upgrading) 10.01 or 10.10, upgrading from 10.0 to 10.01 or 10.10, or upgrading from 10.01 to 10.10, see *Installing HP-UX 10.10 and Updating HP-UX 10.01 to 10.10*.

---

## PCx-Processor Support Pre-Enable

For 10.10, low level kernel enhancements have been made, mostly in the initialization path, to support SPU platforms based on the PCXL2 processor. PCXL2 processors will be recognized by the kernel and appropriate paths through the kernel are taken.

Traps on loads to GR0 will be suppressed, following the PA-RISC 2.0 architectural definition. All platforms conform to the PA-RISC 2.0 semantics in this respect. PCXL2, PCXL, PCXT, and PCXS processors now suppress traps on loads to GR0, just as PCXT' does.

## Configuration

PCXL2 systems require the above changes to boot.

As with PCXT' systems, all other systems will not generate a trap on load to GR0.

---

## Distributed Interrupts and I/O Forwarding

Features:

- For 10.0 systems, the device interrupt load for heavily used disk and networking drivers are “distributed” among all the processors, instead of being forwarded to the monarch processor.
- I/O requests for heavily used disk drivers are now forwarded to the processor scheduled to handle the associated device interrupt.

In pre-10.0 systems, all device interrupts were forwarded to the monarch processor. This could lead to “interrupt saturation” on the monarch on some workloads. That is, the monarch processor could end up processing interrupts most of the time while the other processors remained relatively idle.

In 10.0 systems, the interrupts for some of the most heavily used disk and networking drivers (HPFL, scsi3, GSC SCSI, lan3 and lan6) are “distributed”.

The interrupts corresponding to each instance of one of these drivers are assigned to the next available processor on a round robin basis. Normally this assignment is done automatically, either during I/O system configuration while booting up or on the first I/O operation. The drivers call an I/O services function to accomplish this.

Each instance of one of these drivers corresponds to an IO card. So, the interrupt distribution is on a per card basis and not on a per device basis.

With the introduction of distributed interrupts, an additional performance gain is possible by forwarding an I/O request to the same processor slated to handle the associated device interrupt. This takes advantage of the cache locality of I/O data structures, potentially reducing third-party cache hits and improving MP scaling.

The I/O forwarding feature is currently only implemented for NIO disk devices (HPFL, scsi3, and scsi1). The drivers call an I/O services function defined in the 800 context dependent I/O environment to forward the request to the appropriate processor.

## Impact/Performance

Distributed interrupts may have a beneficial performance impact on some I/O intensive applications. This is especially true as the number of CPUs in the system is increased.

I/O forwarding may provide a performance gain on MP systems that use NIO HPFL and/or SCSI interfaces and disk devices.

For drivers that have not been modified to distribute interrupts, the interrupts will continue to be forwarded to the monarch processor. Likewise, drivers that do not enable I/O forwarding will start up an I/O request on the processor that initially received the request.

## Limitations

The interrupts for various instances of the drivers are assigned to different processors on a round robin basis. This could result in an uneven distribution at times.

The interrupt distribution is on a per card basis. As the number of devices on a card increases, there is a possibility of interrupt saturation occurring on the CPU to which the card's interrupts are assigned.

---

## Logical Volume Manager (LVM)

---

### Note

For all 10.x releases, some LVM commands may not work properly when alternate links (also called alternate paths or pvlincs) are configured to Model 10 (A3231A) or Model 20 (A3232A) High Availability Disk Arrays. A patch correcting those commands will be released for each 10.x release. The patch for 10.10 should be installed after you install or upgrade to 10.10.

The affected LVM commands will only exhibit problems when you try to work with the alternate links. If you do not configure your A3231A or A3232A hardware for alternate links, you will not experience any problems. The *HP 9000 Servers*

*Configuration Guide*, HP Part Number 5963-5318E (March 1995 edition, pages 4-64 to 4-66), lists supported configurations that will work once the patch is installed. If you configure your hardware for alternate links, but do not use LVM to configure those links into the LVM configuration until you install the patch, only the `vgscan` command may have problems. (`vgscan` is used to recover from the loss of the `/etc/lvmtab` file—an error that will not happen under normal operating conditions.)

---

For 10.10, you can create “raw” logical volumes (that is, logical volume that are not used for a file system or swap space).

---

## Shared Logical Volume Manager (SLVM)

For 10.10, LVM has been modified to support volume groups that are connected to multiple systems. This feature is called Shared LVM (SLVM) and is supported only on S800 machines.

SLVM is a mechanism that permits multiple systems in an MC/LockManager cluster to share (read/write) disk resources in the form of volume groups. The objective is a highly available system by providing direct access to disks from multiple nodes and by supporting mirrored disks, thereby eliminating single points of failure.

SLVM permits a two system cluster to have read/write access to a volume group by activating the volume group in shared mode.

SLVM is designed to be used only by specialized distributed applications (such as Oracle Parallel Server) that use raw access to disks, rather than going through a file system. The applications must provide their own concurrency control for their data, as well as transaction logging and recovery facilities, as appropriate. Applications that are not network aware, such as file systems, will not be supported on volume groups activated in shared mode.

SLVM requires services provided by MC/LockManager and thus only clusters that have MC/LockManager will be able to use shared activation.

The following commands have been modified to support SLVM requirements:

### 3-22 Major Changes for HP-UX 10.10



- **vgchange**

The first release of 10.0 supported the “**a -e**” activation mode or exclusive mode. (Exclusive mode permits only one of the two nodes to have read/write access to the volume group the other may have read access but not read/write access.)

For 10.10, **vgchange** has been modified to support the shared activation mode. It now has the “**-c y -S y**” option to mark the attributes of the volume group as shareable and the “**-a s**” option to activate the volume group in shared mode.

- **vgdisplay**

**vgdisplay** will show the current activation mode.

The following commands are not supported for volume groups that are activated in shared mode. You must go to exclusive mode for these operations.

- **lvchange**
- **lvcreate**
- **lvextend**
- **lvlnboot**
- **lvreduce**
- **lvremove**
- **lvrmboot**
- **lvmerge**
- **lvsplit**
- **pvchange**
- **pvmove**
- **vgextend**
- **vgreduce**
- **vgremove**

Note the commands **lvsplit** and **lvmerge** can be done in exclusive mode.

If you need a high availability cluster, you will need to carefully plan your use of shared activation with MC/LockManager.

Shared capabilities are supported only on clusters that use MC/LockManager and only clusters that have two systems.

## Performance

Synchronization of mirrored volume groups will be slower on shared volume groups.

## Compatibility

Once a volume group is marked shared, it retains the shared characteristics. If the volume group is moved to a system without shared capabilities, the shared characteristics must be removed with the “-c n -S n” option of `vgchange`.

## Upgrade from 9.04

To upgrade from the 9\_04 special, you need to bring your cluster down and upgrade each system to 10.0 before bringing the cluster up again.

---

## UNIX95

To make the kernel UNIX95-compliant, the following has been done for 10.10:

- minor changes to existing interfaces
- addition of new interfaces
- signal changes

Performance should not be impacted by any of the following kernel interface changes. However, there might be small degradation in:

- the signal handling routines due to the need to check if `siginfo` has been specified.
- allocating `siginfo` data structures when the process first turns on the `SA_SIGINFO` flag.
- during fork if the parent has specified `SA_SIGINFO`, the child will automatically allocate an additional block of memory for a `siginfo` structure.

## Modified Interfaces

### **fstat(), lstat(), and stat()**

These interfaces get the file status.

*For 10.10:*

- There are no kernel changes.
- Implemented in `libc`.

### **ftruncate() and truncate()**

These interfaces truncate a file to a specified size.

*For 10.10:*

- The kernel changed to return `errno EINVAL` instead of `EFBIG` for a negative file size argument.

### **getitimer() and setitimer()**

These interfaces get and set the value of the interval timer for the process.

*For 10.10:*

- There is no code change for `getitimer`.
- For `libc`, `setitimer` checks for negative `timeval tv_sec`.
- In the header file `<time.h>`, the field `tv_sec` in the definition of `struct timeval` changes from unsigned long to long.
- The `timeval` structure and the `itimerval` structure is visible under an `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.

### **getrlimit() and setrlimit()**

These interfaces return and set the maximum system resource consumption.

*For 10.10:*

- Defines new type `rlim_t` as `int`.
- Increase value of `RLIMIT_NLIMITS`.
- Moves `RLIM*` out of `_INCLUDE_HPUX_SOURCE` namespace and into the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- Implemented in the kernel by adding support for new options:

- RLIMIT\_CPU: maximum cpu time used by a process
- RLIMIT\_AS: maximum amount of a process' address space

### **gettimeofday()**

This interface returns the time in a `timeval` structure.

*For 10.10:*

- Prototype changed to use `void*` instead of the current structure `timezone*`.
- The `tv_sec` field in `struct timeval` changes from unsigned long to long.
- The prototype and `struct timeval` and `struct timezone` are within the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.

### **mmap()**

This interface maps objects into virtual memory.

*For 10.10:*

- Prototype changed to have `void*` return type instead of `caddr_t`.
- The current prototype is still available under `!_XPG4_EXTENDED`.
- In the kernel, for overlapping `mmap()` with `MAP_FIXED` option requests, `errno` will be `EINVAL` instead of `ENOMEM`.

### **mprotect()**

This interface modifies access protections of memory mapping.

*For 10.10:*

- No code changes.
- Prototype change to `void*` from `caddr_t`.
- The current prototype is still available under `!_XPG4_EXTENDED`.

### **msync()**

This interface synchronizes a mapped file.

*For 10.10:*

- No code changes.
- Prototype change to `void*` from `caddr_t`.
- The current prototype is still available under `!_XPG4_EXTENDED`.

## **3-26 Major Changes for HP-UX 10.10**

### **munmap()**

This interface unmaps an object from virtual memory.

*For 10.10:*

- No code changes.
- Prototype change to `void*` from `caddr_t`.
- The current prototype is still available under `!_XPG4_EXTENDED`.

### **readlink()**

This interface returns the contents of the specified symbolic link.

*For 10.10:*

- Prototype is now in the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- The third parameter changes from `int` to `size_t`.

### **readv() and writev()**

These interfaces perform vectored read/write operations on a file.

*For 10.10:*

- No code changes.
- Header file changed to define `size_t`.
- Changed `struct iovec`.
- Changed the prototypes in `<sys/uio.h>`.

### **select()**

This interface performs synchronous I/O multiplexing.

*For 10.10:*

- Prototype changed in `<sys/time.h>` from `int` to `size_t` and `int *` to `void *` (size equivalent).
- Prototype moved out of `_INCLUDE_HPUX_SOURCE` namespace into the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- New code is added to the kernel that returns `EINVAL` if the number of file descriptors is greater than `FD_SETSIZE`.
- Addition of a check for negative time values.

## **setpgrp()**

This interface sets the process group ID and the session ID.

*For 10.10:*

- Prototype is now visible in the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- Implemented in the kernel by modifying the behavior of `setpgrp()` to SVR4. UNIX95 `setpgrp()` should provide a SVR4 type `setpgrp()` system call. The current SVR4 implementation only allows the session to change under the following conditions:
  - The process is not currently a session leader
  - If the process is a process group leader, it must be the only member of the process group.

## **sigignore(), sigrelse(), sighold, sigpause(), and sigset()**

These interfaces set the signal disposition of a process.

*For 10.10:*

- Prototypes for these functions moved from the `_INCLUDE_HPUX_SOURCE` namespace to the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- These functions are copied from `libV3` to `libc`.
- `sigpause(BSD)` is the default behavior. When compiling in the UNIX95 environment, the `sigpause(V3)` behavior is used.
- `sigpause(BSD)` is moved from `libc` to avoid the incompatibility introduced by moving `sigpause(V3)` to `libc`.

## **sigstack()**

This interface sets the signal stack context.

*For 10.10:*

- The field `ss_sp` in the `sigstack` structure is changed from `char *` to `void *`.
- Prototype changes from `const struct sigstack *` to `struct sigstack *`.
- In the kernel, if an attempt is made to modify an active stack, an `errno` of `EPERM` is returned.

### **waitid(), waitpid(), and wait3()**

These interfaces wait for a child process to change state.

*For 10.10:*

- `waitid()` is implemented as a new system call `waitid()`.
- For `waitpid()`, a `WCONTINUED` option is added to the `_WAITMASK` symbolic constant.
- For `wait3()`, the kernel now supports the newly-defined options; the kernel now returns resource usage information.
- Addition of a new flag for the `p_flag` field in the `proc_t` structure `SWCONT` to indicate that the process was stopped and has been continued.
- Addition of a new flag `WCONTFLG` for the corresponding wait return status.
- Replaced the `WSTOPPED` constant in `<sys/wait.h>` with `WSTOPFLG`.

### **New Interfaces**

#### **getcontext() and setcontext()**

These interfaces get or set current user-level context of the calling process.

Features:

- The new `ucontext` structure, `ucontext_t`, and `mcontext` structure, `mcontext_t` is visible under the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- Function prototypes are also visible under the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.
- There are new fields `u_prevcontext` and `u_curstacks` in `struct user`, in `<user.h>`.
- `sigset_t` is moved to be visible under `<signal.h>`.
- Implemented in `libc`, makes new system calls `getcontext()` and `setcontext()`
- References to the old `getcontext` and `setcontext` is changed to `ogetcontext` and `osetcontext`, respectively.

#### **getdtablesize()**

This interface returns the total number of file descriptors currently available for a process.

Features:

- Implemented in `libc`, makes a system call to `getrlimit()`.

### **gethostid()**

This interface returns the identifier of the current host.

Features:

- Implemented in `libc`.
- This interface is not guaranteed to be unique or reproduceable.

### **getpagesize()**

This interface returns the number of bytes in a system page.

Features:

- Implemented in `libc` for portability for older software.
- Makes a system call to `sysconf()`, which is the preferred and faster method of getting the information.

### **getrusage()**

This interface returns the resources utilized by the calling process and its' children.

Features:

- Implemented in `libc` as a stub, makes a system call to `SYS_GETRUSAGE`.
- This functionality is already available, but is currently unused.

### **makecontext() and swapcontext()**

These interface manipulate user-level context for providing user-level context switching between multiple threads of control within a process.

Features:

- These interfaces are implemented in `libc`
- Only available when linking archive with `libc.a`.

## **3-30 Major Changes for HP-UX 10.10**



### **setreuid()**

This interface sets the real and effective user IDs of the current process.

Features:

- Implemented in `libc`.
- Makes a system call to `setresuid()`.

### **sigaltstack()**

This interface sets and/or gets signal alternate stack context.

Features:

- Implemented in `libc`, a new system call to `sigaltstack()` is added to the kernel. It follows the general flow of code from the existing `sigstack()`.
- `u_signalstack` is a new user structure field that replaces the old `u_sigstack` field.
- `<user.h>` is used for both `sigstack` and `sigaltstack` information in the user structure.
- `stack_t` is a `sigaltstack` structure in `<signal.h>` used in the implementation and user interface for `sigaltstack`.
- `stack_t` resides in the `_INCLUDE_XOPEN_SOURCE_EXTENDED` namespace.

### **ualarm()**

This interface sends a `SIGALRM` to the calling process after a specified time.

Features:

- Port of BSD `ualarm()` function to HP-UX.
- Implemented in `libc` by making a system call to `setitimer()`.

### **usleep()**

This interface suspends the calling process for a specified time.

Features:

- Port of BSD `uslepp()` function to HP-UX.
- Implemented in `libc` by making a system call to `setitimer()`.

## utimes()

This interface sets the access and modification times of a specified file.

Features:

- Implemented in `libc` by making a system call to `utime()` while rounding up microsecond arguments to the nearest second for HP-UX.

## Signal Changes

- New `SIGXCPU` signal when CPU time limit is exceeded.
- New `SIGXFSZ` signal when file size limit is exceeded. These two signals push the total number of signals in HP-UX to more than 32. This necessitates design changes in signals support in the kernel and also requires changes to help protect old binaries from being unjustifiably surprised by the new signals.
- New `sigaction` option `SA_RESTART`: the ability to specify at handler installation time whether interrupted signal calls are to be restarted.
- New `sigaction` option `SA_NOCLDWAIT`: the ability to prevent the system from creating zombies for exiting child processes and still specify a signal handler to receive the corresponding `SIGCLD` signals.
- New `sigaction` option `SA_SIGINFO`: the ability to request optional new `siginfo` and `ucontext` structures to be passed to the interrupt handler.
- New `bsd_signal()` routine to provide the historical BSD `signal()` interface for older applications.
- New `siginterrupt()` routine to change the restart behavior when a system call is interrupted by the specified signal.

## Namespace Collision

- Due to namespace collision problems, *all* of the `sigcontext` names (except those defined in UNIX95) must be renamed to have a double underscore (`__`) prefix. However, `#define` redefines the old names to the new names so that applications will not be affected.

---

### Note

New applications should *not* be written with BSD signals because they use 32-bit `ints` as signal masks (such as `sigblock()`), which cannot handle more than 32 signals. Instead, use ISO/IEC-9945 (POSIX 1003-1-1990) signals. If

## 3-32 Major Changes for HP-UX 10.10

you have spare cycles, you may want to convert to POSIX so you can handle SIGXCPU, SIGXFSZ, and future new signals.

---

## Compatibility

The table below summarizes the backwards compatibility of the changes described in this section. Following the table is a listing showing the previous and current 10.10 structures in `signal.h`.

COMMAND	COMPATIBLE?	ADDITIONAL INFORMATION
fstat() lstat() stat()	Yes	
truncate() ftruncate()	Yes	If the file size is a negative value, EINVAL is returned. Previously, there was no error return value, but you should still check for additional error conditions.
getcontext() setcontext()	Yes	These are new commands, except there is an obsoleted getcontext call. The previous call has been renamed, so will not be an issue.
getdtablesize()	Yes	New command -- no compatibility issues.
gethostid()	Yes	New command -- no compatibility issues.
getitimer() setitimer()	Yes	The type of tv_sec in the timeval structure is changed from unsigned long to long. Because time should be a positive value, this should not have an impact. EINVAL is returned for negative times passed into setitimer. There is no source compatibility issues because the old timeval structure is placed inside an <code>_INCLUDE_XOPEN_SOURCE_EXTENDED</code> block.
getpagesize()	Yes	New command -- no compatibility issues.
getrlimit() setrlimit()	Yes	Adding new RLIMIT_CPU and RLIMIT_AS options. Also, increasing RLIMIT_NLIMITS to 8.
getrusage()	Yes	New command -- no compatibility issues.
gettimeofday()	Yes	The prototype is changing to use void*

instead of the current struct timezone\*.  
 The tv\_set field in struct timeval is  
 changing from unsigned long to long. There  
 is no source compatibility issues because the  
 prototype, struct timeval, and struct  
 timezone is placed inside an `_INCLUDE_XOPEN_  
 SOURCE_EXTENDED` block.

<code>makecontext()</code> <code>swapcontext()</code>	Yes	New command -- no compatibility issues.
<code>mmap()</code>	Binary compatible; not source compatible	For overlapping <code>mmap()</code> requests, <code>errno</code> is <code>EINVAL</code> instead of <code>ENOMEM</code> . Also, the prototype is changed to <code>void*</code> return type from <code>caddr_t</code> , but the prototype will be under <code>ifdef</code> .
<code>mprotect()</code>	Yes	The prototype is changed to <code>caddr_t</code> return type from <code>void*</code> . The prototype will be under <code>ifdef</code> , so there is source compatibility.
<code>msync()</code>	Yes	The prototype is changed to <code>caddr_t</code> return type from <code>void*</code> . The prototype will be under <code>ifdef</code> , so there is source compatibility.
<code>munmap()</code>	Yes	The prototype is changed to <code>caddr_t</code> return type from <code>void*</code> . The prototype will be under <code>ifdef</code> , so there is source compatibility.
<code>readlink()</code>	Yes	The prototype is now visible in the <code>_INCLUDE_XOPEN_SOURCE_EXTENDED</code> namespace.
<code>read()</code> <code>writev()</code>	Yes	Changes were made to: header files to define <code>size_t</code> , <code>struct iovec</code> , and prototypes in <code>&lt;sys/uio.h&gt;</code> . The prototype will be under <code>ifdef</code> , so there is source compatibility.
<code>select()</code>	No	Is not binary or source compatible; the command has been changed to handle a negative <code>tv_usec</code> , which should rarely be needed. Also, the error return when the number of <code>fds</code> is greater than <code>FD_SETSIZE</code> is changed to return <code>EINVAL</code> .
<code>setpgrp()</code>	No	Will display SVR4 behavior; SVR4 implementations only allows the session to be changed under these conditions: * The process is not currently a session leader * If the process is a process group leader, it must be the only member of the process

### 3-34 Major Changes for HP-UX 10.10

		group. Existing code does not change the session ID.
-----		
setreuid()	Yes	New command -- no compatibility issues.
-----		
sigaltstack()	Yes	New command -- no compatibility issues. u_signalstack is a new field replacing the old u_sigstack field in the user structure <user.h>. stack_t is a new sigaltstack structure in <signal.h>.
-----		
sigignore() sigrelse() sighold() sigpause() sigset()	Yes	The prototypes for these functions is moved from the _INCLUDE_HPUX_SOURCE namespace to within the _INCLUDE_XOPEN_SOURCE_EXTENDED namespace. sigpause(BSD) is moved from libc to accommodate the V3 version being moved into libc.
-----		
sigstack()	Yes	The kernel now returns a new error EPERM if an attempt is made to modify an active stack. This is not a compatibility issue because all applications should always check for additional system-specific error conditions. Also, the field ss_sp in the sigstack structure is being changed from char* to void*. The prototype is changing from const struct sigstack* to struct sigstack*. The prototype will be under ifdef, so there is source compatibility.
-----		
ualarm()	Yes	New command -- no compatibility issues.
-----		
usleep()	Yes	New command -- no compatibility issues.
-----		
utimes()	Yes	New command -- no compatibility issues.
-----		
waitid() waitpid() wait3()	Binary compatible; not source compatible	The WSTOPPED constant is being replaced with the WSTOPFLG constant so that WSTOPPED can be used as an option flag, as described in UNIX95.  An application using WSTOPPED will have to change to use the supported macros (such as WIFSTOPPED) defined in <sys/wait.h>. Also, existing programs that do not pass in NULL as the third argument to wait3() cannot expect an error to be returned.
-----		
New signals	No	Two new signals (SIGXCPU and SIGXFSZ) have

been added. The effects on compatibility are:

- \* Most applications should still run because UNIX95 processes and non-UNIX95 processes are handled differently. Specifically, the system will not send a system-generated SIGXCPU or SIGXFSZ to an old process, except when an older process has indicated awareness of the SIGXCPU signal (that is, the process blocks ignores or catches a new signal).
- \* The signal mask is now longer than 32 bits. Any program accessing the signal mask (sc\_mask) in the sigcontext structure (the third argument passed into the signal handler) will not work. sc\_mask is a sigset\_t (an eight-word structure) rather than a single long. The offset of the sc\_mask field in the sigcontext structure will also change. You must make corresponding changes in your code to reflect the new sc\_mask. Use sigprocmask() instead of sigsetmask() to change a process' signal mask. The former accepts a signal mask that is not of type sigset\_t.
- \* Any program that uses the  
(sp - (sizeof sigcontext))  
method to locate a sigcontext pointer will not work. Change your code to use the fm\_psp field in the frame\_marker. The fm\_psp field is located at  
(sp - sizeof(char \*)), where sp is the value of the stack pointer on entry to the sigcontext structure. By making this change, your code will no longer depend on the size of sigcontext.
- \* Old programs using BSD signals may be killed if you issues a kill (SIGXCPU) command to the process. (This is no worse than kill(-9), which is unprotectable.) The system will not do this because it recognizes that the process cannot handle SIGXCPU.
- \* The system cannot differentiate between old and new shell scripts; the system can only distinguish between the interpreters (such as sh). Some system scripts may not know how to process the two new signals.

=====

The old structures in `signal.h` are as follows:

```
    struct siglocal_misc {
int sm_syscall; /* interrupted system call if any */
int sm_onstack; /* sigstack state to restore */
int sm_mask; /* signal mask to restore */
char sm_syscall_action; /* what to do after sys call */
char sm_eosys;
unsigned short sm_error;
int sm_rval1;
int sm_rval2;
int sm_arg[NUMARGREGS];
    };

    struct siglocal {
struct siglocal_misc sl_misc; /* miscellaneous signal state */
struct save_state sl_ss; /* user saved state */
    };

    struct sigcontext {
struct siglocal sc_sl; /* local frame containing context */
int sc_args[NUMARGREGS]; /* arguments to handler */
struct frame_marker sc_sfm;
    };
```

The new structures in `signal.h` are as follows:

```
    struct siglocal_misc {
int sm_syscall; /* as before */
int sm_onstack; /* as before */
int sm_reserved; /* old location of sm_mask */
char sm_syscall_action; /* as before */
char sm_eosys; /* as before */
unsigned short sm_error; /* as before */
int sm_rval1; /* as before */
int sm_rval2; /* as before */
int sm_arg[NUMARGREGS];
    };

    typedef struct save_state mcontext_t;

    struct ucontext {
mcontext_t uc_mcontext; /* user saved state */
sigset_t uc_mask; /* 8 word signal mask */
... /* other ucontext fields */
    };
```

```

    struct sigcontext {
    struct siglocal_misc sc_slm; /* same save for sm_mask */
    struct ucontext sc_uc; /* new. */
    struct siginfo sc_si; /* new. siginfo */
    int sc_args[NUMARGREGS]; /* register spill, old. */
    struct frame_marker sc_sfm; /* frame, old. */
    };

#define sc_mask sc_uc.uc_mask

typedef struct sigaltstack { /* Used with sigaltstack(). */
    void *ss_sp; /* specifies new base of signal stack */
    int ss_flags; /* specifies the new stack state */
    int ss_size; /* specifies the new stack size */
} stack_t;

<user.h> user structure field:
struct {
    void *uss_alternatesp;
    unsigned int uss_size;
    unsigned int uss_enabled:1;
    unsigned int uss_inuse:1;
} u_signalstack;

```

## UNIX95 Commands/Libraries

HP-UX commands and libraries conform to X/Open Portability Guide Issue 4, Version 2, also known as UNIX95.

See the section “Library Routines (libc)” in Chapter 6 for details on specific changes within each `libc` interface or command. For a complete list of UNIX95 commands and interfaces, consult the appropriate “X/Open CAE Specification, Issue 4, Version 2” publication or *Going Solo* published by Prentice Hall.

*Going Solo* also includes a CD ROM that contains the above specification in addition to other UNIX95 manpages.

To enable full UNIX95 behavior, do the following:

- Set the environment variable `UNIX95`. For example, for the POSIX `ksh` shell:

```
export UNIX95=
```

- Pre-pend the path `/usr/xpg4/bin` to the `$PATH`. For example, for the POSIX `ksh` shell:

### 3-38 Major Changes for HP-UX 10.10



```
export PATH="/usr/xpg4/bin:$PATH"
```

To build UNIX95- conformant applications, follow the above steps, in addition to the following:

- Define the compile option `_XOPEN_SOURCE_EXTENDED=1`. See the `cc` manpages for more details.

Some UNIX95 behavior is not binary-compatible with previous HP-UX behavior. Thus, it is advisable to enable full UNIX95 behavior only in applications that have been verified to work in the UNIX95 environment. If an application has not been verified to work in the UNIX95 environment or if compatibility with previous HP-UX behavior is desired, it is not recommended to enable the full UNIX95 behavior as described above.

---

## HP Common Desktop Runtime Environment (CDE 1.0)

CDE is the new industry standard UNIX desktop. CDE provides these standard desktop applications:

- Help System
- Front Panel
- Mailer
- Calendar
- Style Manager
- File Manager
- Application Manager
- Print Manager
- Text Editor

The design of the CDE desktop incorporates and enhances many HP VUE features. Although the CDE desktop has a similar appearance to HP VUE, there are important differences. The differences include:

- New and more customizable Front Panel
- Graphical MIME-enabled mail application
- Graphical Calendar
- Graphical Print Queue Manager

- New terminal emulator
- Action and datatype syntax changes
- ToolTalk messaging support
- Desktop application registration

## **Development Environment**

The CDE development environment provides an application programming interface for many of the desktop components.

## **Printing and Building Help Volumes in Multibyte Locales**

The user's LANG environment variable must match the locale specified for a given help volume to print or build the volume correctly.

The locale associated with a help volume is set using the `LanguageElementDefaultLocale` entity declaration, or using a combination of the `LanguageElementDefaultLocale` and `LanguageElementDefaultCharset` entity declarations.

If the locale is not specified for a help volume, the value of the LANG environment variable is used.

## **CDE Migration Tool**

The VUEtoCDE application provides a way to migrate personal and system-wide HP VUE customizations to the CDE desktop. VUEtoCDE is included with the HP-UX 10.10 operating system.

When upgrading your HP VUE environment to CDE, install CDE and the VUEtoCDE product from the operating system media. The VUEtoCDE application is installed in the `/opt/Migration` directory.

Refer to *HP CDE Getting Started Guide* (B1171-90104) for instructions for migrating HP VUE customizations to CDE using VUEtoCDE.

## Accessing Font Aliases for Remote CDE Sessions

Your CDE session must have access to CDE font aliases to run properly. Typically, these aliases are installed with the fonts provided with your X server.

However, when running CDE as a remote session using the command

```
X -query... fp+ tcp/hostname:7000
```

or running CDE from an X terminal, it is possible that the font aliases may not be available to the X server.

This can happen when the fonts on the server are from a prior release of HP-UX or from another vendor. In this situation, you may see fonts that do not work correctly with the local language you have selected. Fortunately, CDE provides “backup” CDE font aliases that your X server can use. These are located in the directory

```
/usr/dt/config/xfonts/locale name
```

This location must also contain a `fonts.dir` file. This file is required by the X server to use the font aliases mechanism.

If you are running CDE remotely, your system administrator can cause the CDE font aliases to be used by setting up a font server to run on the system hosting CDE, making that font server use the CDE font aliases for the language you choose.

For example, if you want to work in Japanese EUC (locale name `ja_JP.eucJP`), a font server should be started on the CDE system using this command:

```
/usr/bin/X11/fs -port 7000 -daemon -quiet_if_addrinuse
```

In the font server’s configuration file (the default configuration file is `/etc/X11/fs/config`), make sure the CDE font alias directory is contained in the catalogue definition. For example, to support a user running CDE in Japanese EUC, the font server’s configuration file might contain the following:

```
catalogue = ...,/usr/dt/config/xfonts/ja_JP.eucJP/
```

Alternatively, your system administrator can copy the `fonts.alias` files from the CDE host system to your local system and incorporate them into your X server's `fonts.alias` file.

## Changing Languages Between Sessions in CDE

When you log into the CDE desktop, you can select the language (or locale) that you want to use for your session. A list of languages is displayed when you click the Option button in the login screen.

If you change the language you are using between CDE sessions, previously saved local language customizations may conflict with your new language session.

For example, suppose you log in with language `ja_JP.eucJP` and create Japanese workspace names. You then save the session when you exit the desktop. Suppose the next time you log in, instead of choosing `ja_JP.eucJP`, you select German, `de_DE.iso88591`. The desktop will attempt to display the Japanese workspace names using fonts appropriate for the German language. The workspace names will be unreadable.

This conflict results only if you change between two languages that use different coded character sets. You can change from either C or English locale without difficulty. Likewise, you can switch between similar locales, such as French and German, which are both Western European locales.

Any personal customizations that contain local language characters can cause similar symptoms as that described above. Personal session customizations are saved in your `$HOME/.dt` directory.

If you want to select a language that is different from your previously saved session, you should first start CDE in a fail safe session. From the terminal window provided, remove the files that contain local language text.

Files that might contain local language characters include:

- `$HOME/.dt/session/current/dt.resources`
- `$HOME/.dt/help`
- `$HOME/.dt/types/*`

If you are not sure what files contain local language characters, remove the directory `$HOME/.dt/session` using this command:

### 3-42 Major Changes for HP-UX 10.10

```
rm -r $HOME/.dt/session
```

This causes a default session to be invoked when you log in with the new language.

In addition, you should remove any help files accessed with the prior local language by executing:

```
rm -r $HOME/.dt/help
```

Labels and comments in action definitions can also include localized data. If so, the action labels need to be modified to be readable in the new language selected for your CDE session. User-defined actions are located in `$HOME/.dt/types/*`.

---

<b>Note</b>	If you delete your personal customization directory, <code>\$HOME/.dt/*</code> , you can avoid the problems that occur when changing languages. However, remember that you will lose any actions or other customizations you might have created. Removing the entire directory is therefore not recommended.
-------------	--

---

## HP CDE and Commercial Security (Trusted Systems)

Commercial security is not supported by CDE. HP CDE uses several components to authenticate users when they log into the system. However, not all of these components utilize the trusted system security features.

If a system is configured to start up CDE automatically, you cannot convert it to a trusted system without disabling CDE. If you are using SAM (System Administration Manager) to enable commercial security on your system, an error message will be displayed if CDE has been enabled.

Refer to the `dtconfig(1)` manpage for information about configuring CDE on your local system. Note that `dtconfig` does not allow CDE to be configured on trusted system.

## **CDE Limitations**

The CDE limitations are as follows:

- CDE Application Builder is not included in this release. It is an optional component of the CDE X/Open standard.
- Commercial security is not supported.
- SharedPrint is not integrated into CDE.
- Drag and drop of ImageView and Audio files into a CDE window is not supported.
- Files cannot be printed directly from ImageView.

### **Addition to** *Advanced User's and System Administrator's Guide*

"Creating an Action that Runs as a Different User" in Chapter 10 has this addition:

When invoking an action that runs as a different user, make sure that the specified user has access to all resources required to complete the action. This includes, but is not limited to, the following:

- X display access (use `xauth`, `xhost`, `acl`'s or `mode` on `.Xauthority`, and so on).
- File permissions (use `chmod`, `acl`'s, and so on).
- NFS access (grant local `root` access to NFS-mounted file systems).
- Environment variables (modify `$HOME`, `$PATH`, and so on, for the new user if necessary)

---

## NIKE Array Enhancements

For 10.10, the enhancements to the NIKE array are:

- New 2Gb LP 7200 rpm Disk Drive
- New 4Gb LP 7200 rpm Disk Drive
- Flare 8.X Firmware

Features:

- Read and write cache on each SP—this enables single SP with Cache.
- Read-ahead caching—this improves SP performance.
- Optimization of RAID 1 and RAID 1/0—this increases RAID 1 and 1/0 performance.

---

## 60K File Descriptors

As of 10.10, the maximum allowed number of simultaneous open files, or maximum number of file descriptors, has been raised from 2048 to 60000 per process. This value is specified by the constant `MAXFUPLIM` (and the equivalents `_MAXFUPLIM` and `FD_SETSIZE`).

This feature is optional because it will not be often used and to minimize the impact on existing code. An application requiring more than 2048 file descriptors must be recompiled with the new symbol `_USE_BIG_FDS` defined. To do this, add the flag `D_USE_BIG_FDS` to the compile command in the application's `makefile`. Or, the symbol can be defined at the beginning of every application source file (via `#define _USE_BIG_FDS`).

As in previous releases, there are also per-application resource limits, independent of the `MAXFUPLIM` limit. To use a large amount of file descriptors, your application might need to modify its `RLIMIT_NOFILE` resource limits with `getrlimit(2)` and `setrlimit(2)`. See the manpages for these system calls for more information. Alternatively, the system-wide `RLIMIT_NOFILE` defaults can be changed (as in previous releases) by modifying the kernel tunables `maxfiles` and `maxfiles_lim`; see the SAM on-line kernel configuration help for more information.

Any machine running an application that uses a large amount of file descriptors might need to be reconfigured with a larger value for the kernel tunable `nfile`. This tunable specifies the per-machine (as opposed to per-process) maximum number of simultaneous open files and is by default much less than 60000. See the SAM online kernel configuration help for more information.

Use of this new feature is not compatible with all libraries. Users need to review the libraries used by their application to determine if they can use this feature. See the section “Impacts” below for more information.

You can use SAM to implement this feature. The kernel tunables table reflect new maximum values for the tunables `maxfiles` and `maxfiles_lim`, from 2048 to 60000.

## Impacts

This new functionality might have three possible affects:

1. *For all user-space code (commands, libraries, applications, and user code):*

User-space code that raises its `RLIMIT_NOFILE` resource limit with `setrlimit` (see above) beyond `MAXFUPLIM` might fail.

The `RLIMIT_NOFILE` resource limit cannot exceed `MAXFUPLIM`. In previous releases, the kernel and user space `MAXFUPLIM` values were the same. To support 60K file descriptors, the kernel `MAXFUPLIM` is now 60000. However, to maintain compatibility with previous releases, user applications by default will still use a `MAXFUPLIM` of 2048. If an application with a small `MAXFUPLIM` “succeeds” in raising its resource limit above 2048, subsequent failures might occur. For example, you can now open more than 2048 files (assuming that the kernel tunable `nfile` is sufficiently large), though `select(2)` data structures are sized for only 2048 file descriptors. A `select` call might hang or might cause a segmentation violation depending on the arguments passed to it. See the `select` manpage and the `<sys/types.h>` header file for more information.

There are no workarounds for this impact; impacted code must be modified.

2. *For code opting for 60K file descriptor functionality:*

The use of 60K file descriptors in conjunction with some libraries is not supported. A library compiled with the old `MAXFUPLIM` value might not work



with applications using the new `MAXFUPLIM` value. Any calls to `select(2)` made by a library function on behalf of the application would fail if you had more than 2048 open files, as in the preceding item above.

The use of 60K file descriptors in conjunction with the following libraries is not supported:

- X Window System libraries
- dce threads libraries
- Pascal and Fortran libraries
- pre-10.10 libraries
- 3D graphics libraries (Starbase, HP PEX, and HP-PHIGS)
- third-party libraries

3. *For code opting for 60K file descriptor functionality:*

Programs that opt for 60K file descriptors might experience excessive memory usage or performance problems.

The `fd_set` data structure used in `select(2)` is sized with the `FD_SETSIZE` constant (equivalent to `MAXFUPLIM`). Increasing this constant from 2048 to 60000 increases the amount of memory used by an `fd_set` from 256 bytes to about 7500 bytes. Most applications that use `fd_sets` use only a few. However, there might be applications that use many. Recompiling those applications to use 60K file descriptors will cause a large increase in memory usage, which may be problematic. The same problem exists for applications that use `MAXFUPLIM` or `FD_SETSIZE` to size other data structures.

For similar reasons, applications opting for 60K file descriptors might experience performance degradation if the application uses `MAXFUPLIM` or `FD_SETSIZE` as a loop bound or as a `select nfd`s argument (see the `select` manpage for details).

## Alternatives

To minimize the impact on existing code, the 60K file descriptor feature is optional. Existing code, even as it is recompiled, continues to use the old `MAXFUPLIM` value of 2048 unless a different value is selected by defining `_USE_BIG_FDS`.

Whether code is recompiled with or without `_USE_BIG_FDS` or even if the code is not recompiled (such as existing binaries), it may be vulnerable to the first impact described above. There is no workaround for this impact.

## Size Requirement

Approximately 1 KB of memory is used by the kernel to store 32 file descriptors, both in the per-process file descriptor table and the system-wide file table (sized by `nfile`, above). 60K file descriptors can use as much as 2 MB. Thus, excessive use of this feature may not be possible on small-memory machines.

---

## Dynamic Host Configuration Protocol (DHCP)

The DHCP Server offers advanced IP address management for TCP/IP LAN computing environments. By automating IP address allocation, the server offers a level of automation not available with today's client-server bootstrap protocol. DHCP preserves your investment in BOOTP by supporting new and existing BOOTP clients, as well as new DHCP clients. DHCP further improves upon previous versions of BOOTP by supporting more configuration options.

Examples of DHCP clients include TCP/IP network printers, X terminals, and Microsoft Windows machines.

The DHCP server can be administered through SAM or manually. Through SAM you can:

- configure DHCP groups for automated IP address allocation
- reserve a fixed IP address for select IP devices
- manage IP address leases
- administer existing or new BOOTP devices
- configure the DHCP/BOOTP client proxy service

There is a new command line tool `dhcptools(1M)` that further simplifies the management of DHCP devices. `dhcptools` provides access to DHCP-related operations for the `bootpd` server. The options provide control for dumping internal data structures, generating a hosts file, previewing client address

assignment, reclaiming unused addresses, tracing packets, and validating configuration files.

DHCP is fully backwards compatible with BOOTP; no changes are required by existing users of BOOTP. You need to configure `/etc/dhcptab` to activate DHCP functionality.

See the `bootpd(1m)` manpage for details.

---

## Additional Tools

Additional tools for 10.10 include the following:

- graphical mailer
- graphical calendar
- Athena Widget library (`Xaw`)
- X Miscellaneous Utilities library (`Xmu`)
- `Imake`
- Desktop Login in the absence of a HOME directory

The graphical mailer and calendar components are provided through the CDE desktop, which includes `dtmail` and `dtcm` applications.

Archived versions of the `Xaw` and `Xmu` libraries are provided with the 10.10 development environment as “contrib” software, along with appropriate header files. The availability of these libraries allow you to compile various applications that depend on these libraries which originate from the X Consortium.

The `Imake` utility is a build tool that was popularized as a result of its use with X Consortium source deliveries. `Imake` is a supported component in the 10.10 development environment.

Another tool is the ability for you to log onto a system when your HOME directory is not present on the system. This is a common occurrence in a networked environment when user HOME directories are NFS mounted. The login interfaces for both VUE and CDE desktop products in 10.10 allows you to be validated and subsequently provided with a “failsafe” session.

---

## VME Services

VME Services, added as a patch after 10.0, is upgraded to include the following:

- Location Monitor (Series 743 and 748 only)
- VME FIFO (Series 743 and 748 only)
- New API functions and `ioctl`s to support the Location Monitor and VME FIFO.

See the following documents for more information:

- *HP-UX 10.0 VME Configuration Guide*
- *HP-UX Driver Development Guide: HP 9000 Series 700 Computers*— “VME Driver Development” chapter and appendix
- *HP-UX Driver Development Reference: HP 9000 Series 700 Computers*

The Location Monitor, VME-specific and 743/748-only, is designed to interrupt a processor whenever a particular range of addresses is read or written to. Multiple processors can watch the same address range and they all can be interrupted if that range is reached.

Within the VME support of the Location Monitor, there are three different methods of notification that the Location Monitor has been triggered:

1. sleep; wake when received
2. poll; return the trigger count since the last poll
3. signal (or interrupt function); asynchronous awake on trigger

The Location Monitor is supported by `vme2` driver `ioctl` calls. New user access data structures are defined in `vme2.h`.

The VME FIFO, VME-specific and 743/748-only, is an 8-bit wide by 16-byte deep hardware shift register. It is read and written from a single address that can be assigned in any modifier space. The primary objective of the VME FIFO is to provide a method of ordering asynchronous communications received by a VME processor.

Whenever a VME FIFO is written into, an interrupt occurs on that processor. If the FIFO is filled, no additional data is permitted and an overflow flag is set. Each FIFO can only be read by the processor it is on.

All 9.05 VME drivers will have to be rewritten for 10.0. No changes are required for 10.01 drivers to work with 10.10.

---

## Large Buffer Cache

The buffer cache can now occupy as much free memory as is available, up to the system limit of 3.75 GB. Prior to 10.10, the implementation of the buffer cache silently limited the size to 1 GB.

The kernel parameters `dbc_min_pct` or `bufpages` can be set beyond 1 GB if that physical memory is available.

This change allows the storage of nearly 3.75 GB of file data in memory. This should produce a noticeable increase of speed for applications that access between 1 and 3.75 GB of file system data. There is one exception: NFS server code is unable to take advantage of a buffer cache over 1 GB due to limitations in the networking drivers.

If the kernel configuration requested, but previously did not receive, a buffer cache larger than 1 GB, the increase in size at 10.10 might change performance and response time, depending on the workload.

---

## pstat(2)

The `pstat(2)` interface has been enhanced to return information concerning two new system contexts:

- process virtual memory information
- system stable storage information

Two new `pstat(2)` interface requests have been added to return the above contexts:

- `pstat_getprocvm()`
- `pstat_getstable()`

`pstat_getprocvm()` returns information specific to a particular process' address space. Detailed information on each process' text, data, stack, address regions, and so on, can be obtained.

`pstat_getstable()` returns information contained in the system's stable storage area. Information such as the system's primary boot path, console/display path, and so on, can be obtained.

Refer to the `pstat(2)` manpage and `<sys/pstat.h>` for details concerning the above interfaces.

All existing `pstat(2)` interfaces are backwards compatible with application binaries.

---

## Remote Procedure Call (RPC) Interfaces

### `<rpcsvc/*.h>`

For 10.10, ANSI C and C++ support to the header files in the directory `/usr/include/rpcsvc` are added. The files affected are:

<code>klm_prot.h</code>	<code>rex.h</code>	<code>rwall.h</code>	<code>ypclnt.h</code>
<code>lock_prot.h</code>	<code>rquota.h</code>	<code>sm_inter.h</code>	<code>yppasswd.h</code>
<code>mount.h</code>	<code>rstat.h</code>	<code>spray.h</code>	<code>ypupdate_prot.h</code>
<code>nlm_prot.h</code>	<code>rusers.h</code>	<code>yp_prot.h</code>	<code>ypvi_prot.h</code>

If you are not using ANSI C or C++, there is little or no impact. For ANSI C and C++ users, you no longer have to modify the include files.

### `rpc_control()`

At 10.10, support for `rpc_control` options are added:

<code>CLGET_FD</code>	Returns the file descriptor for the connection
<code>CLSET_NCLOSE</code>	Causes <code>clnt_destroy()</code> not to close the socket

`CLSET_CLOSE` Causes `clnt_destroy()` to close the socket; this is the default.

## RPC 60K File Descriptors

For 10.10, there is now support for 60K file descriptors. Therefore, you can open 60K minus 1 file descriptors and use RPC. See the section “60K File Descriptors” in this chapter for details.

## Name Service Switch Support for RPC Lookups

The RPC database lookup routines `setrpcent()`, `getrpcent()`, `endrpcent()`, `getrpcbyname()`, and `getrpcbynumber()` now supports the Name Service Switch. See the section “Name Service Switch” in Chapter 5 for details.

---

## TCP/IP Transport

For 10.10, the default TCP Maximum Segment Size (`TCP_MSS`) has been changed from 512 bytes to 536 bytes for consistency with RFC 1122.

---

## Cold Install

For 10.10, the following is available:

- Configuring non-root file systems greater than 4Gb.
- On systems with multiple LAN interfaces, you can now select which interface should be used during the install. The interface chosen will also be the one configured by `set_parms` at system bootup.
- The mini-system tar archive (`/usr/lib/sw/hpux.install/syscmds`) is now compressed using `gzip(1)` instead of the `compress(1)` command. This applies to the `instcmds` archive as well.
- Systems with two disks, where the root disk is less than 600Mb, will by default be configured using LVM with both disks in the root volume group. If the second disk has data you do not want destroyed, modify the configuration to not use that disk during the install.

---

## Software Distributor (SD)

For 10.10, the new `swgettools` command handles the period during an OS upgrade when the SD commands existing on the system are executing with new, and potentially incompatible, shared libraries, but still on the old kernel.

### Features

#### ■ Improved usability—

The default software view has been changed from showing only bundles to showing all top-level software (products and bundles). This allows users easier access to non-HP products that are not shipped in bundles.

#### ■ Reduced log file verbosity—

Previous versions of SD produced excessive amounts of log file and screen messages. The 10.10 version will, by default, produce fewer messages. The previous volume of messages can be achieved by setting command line options.

#### ■ Increase communications reliability—

SD now uses TCP as the default communications protocol for all communications between remote systems. Using TCP increases communications reliability for all users. This does not affect command interface or output.

#### ■ Support for file systems up to 128 GB—

SD fully supports the 10.10 file system enhancements.

#### ■ Assignment of unique numeric identifies to log messages—

The addition of a `log_msgid` option that will have a range of [0..4] to indicate which messages will have an ID number prepended. The default value is 0 (the same as before).

#### ■ Improved bundles removal checking—

Filesets that are part of a bundle that is being removed will only be removed if they are *not* also part of any other valid bundle that is installed on the system or contained in the depot. If the fileset is part of another valid product installed on the system or contained in the depot, it will be removed.



■ **Automatic codeword distribution to clients installing from central depot—**

When a central server is serving a codeword protected CD-ROM to other hosts, it will distribute the codeword entry to those hosts to eliminate the need for system administrators to enter it at each location.

**Impact**

Using `swgettools` is a necessary step in the OS upgrade process for all users.

Usability impacts all users because any user scripts that depend on 10.01 output will need to invoke the SD commands with additional options to cause 10.01 output.

Any scripts that were dependent on the 10.01 default `swlist` output will need to be updated to accommodate changes in the default software view.

You will have functionality automatically removed from the kernel rather than having to reconfigure and rebuild the kernel manually after software removal succeeds. The system will automatically reboot when removing a kernel fileset.

**Obsoleted Series 700 Desktop bundles**

The Series 700 Desktop HP-UX configurations are obsoleted. The bundles affected are:

HPUXDanDT700	Danish HP-UX Desktop Environment
HPUXDutDT700	Dutch HP-UX Desktop Environment
HPUXEngDT700	English HP-UX Desktop Environment
HPUXFinDT700	Finnish HP-UX Desktop Environment
HPUXFreDT700	French HP-UX Desktop Environment
HPUXGerDT700	German HP-UX Desktop Environment
HPUXItaDT700	Italian HP-UX Desktop Environment
HPUXJpnDT700	Japanese HP-UX Desktop Environment with JSE
HPUXKorDT700	Korean HP-UX Desktop Environment with KSE
HPUXNorDT700	Norwegian HP-UX Desktop Environment

HPUXSchDT700	Simplified Chinese HP-UX Desktop Environment with SSE
HPUXSpaDT700	Spanish HP-UX Desktop Environment
HPUXSweDT700	Swedish HP-UX Desktop Environment
HPUXTChDT700	Traditional Chinese HP-UX Desktop Environment with TSE

### **New CDE Bundles for Series 700**

HPUXEngCR700	English HP-UX CDE Runtime Environment
HPUXFreCR700	French HP-UX CDE Runtime Environment
HPUXGerCR700	German HP-UX CDE Runtime Environment
HPUXItaCR700	Italian HP-UX CDE Runtime Environment
HPUXJpnCR700	Japanese HP-UX CDE Runtime Environment with JSE
HPUXKorCR700	Korean HP-UX CDE Runtime Environment with KSE
HPUXSchCR700	Simplified Chinese HP-UX CDE Runtime Environment with SSE
HPUXSpaCR700	Spanish HP-UX CDE Runtime Environment
HPUXSweCR700	Swedish HP-UX CDE Runtime Environment
HPUXTChCR700	Traditional Chinese HP-UX CDE Runtime Environment with TSE

### **New CDE Bundles for Series 800**

HPUXEngCR800	English HP-UX CDE Runtime Environment
HPUXFreCR800	French HP-UX CDE Runtime Environment
HPUXGerCR800	German HP-UX CDE Runtime Environment
HPUXItaCR800	Italian HP-UX CDE Runtime Environment
HPUXJpnCR800	Japanese HP-UX CDE Runtime Environment with JSE
HPUXKorCR800	Korean HP-UX CDE Runtime Environment with KSE

### **3-56 Major Changes for HP-UX 10.10**

HPUXSchCR800	Simplified Chinese HP-UX CDE Runtime Environment with SSE
HPUXSpaCR800	Spanish HP-UX CDE Runtime Environment
HPUXSweCR800	Swedish HP-UX CDE Runtime Environment
HPUXTChCR800	Traditional Chinese HP-UX CDE Runtime Environment with TSE

### **New Core Bundles for Series 700**

The following new bundles will be included on the Series 700 HP-UX Core media:

KernDevKit700	HPUX Kernel Developers Kit
DCESystemAdmin	DCE System Administration Utilities
LiteConfig	Small Memory (Desktop) Configuration

### **New Core Bundles for Series 800**

The following new bundles will be included on the Series 800 HP-UX Core media:

KernDevKit800	HPUX Kernel Developers Kit
DCESystemAdmin	DCE System Administration Utilities

---

## **Known Problems**

The Software Release Bulletin (SRB), containing information on known problems that have been fixed in 10.10, is located on the 10.10 media in `/usr/share/doc/10.10SRB`.



# 4

## Major Changes for HP-UX 10.0 and 10.01

---

This chapter covers the following topics:

- New Features of HP-UX 10.0 and 10.01
- Understanding the Changes
- Support Information
- New Commands
- Description of the New Features:
  - The HP-UX 10.0 File System Layout
  - NFS Diskless
  - I/O Convergence
  - Kernel Convergence
  - Software Distributor
- Known Problems and Workarounds

The theme of HP-UX 10.0 version B.10.01 (referred to as “10.01” in this document) is interoperability. This release not only erases most of the inconsistencies between the Series 700 and 800 platforms, but also lays the foundation for a truly vendor-independent operating environment.

References to “10.x” refer to both HP-UX 10.0 and 10.01.

---

## New Features of HP-UX 10.0 and 10.01

New at HP-UX 10.0:

- A new file system layout based on the industry standard, AT&T SVR4, often called simply “V.4.” This layout was implemented at HP-UX 10.0, and has not changed for 10.01.

It is important to understand that the V.4 layout is a new *file system layout*, not a new *operating system*. (See “The HP-UX 10.0 File System Layout”.)

- Convergence between HP’s Series 700 and Series 800 computers.
  - I/O subsystem. (See “I/O Convergence”.)
  - Kernel configuration. (See “Kernel Convergence”.)
- A new tool for installing and updating software. (See “Software Distributor”.)

New at HP-UX 10.01:

- Support for 9.x to 10.01 upgrade.
- A new diskless protocol based on the industry standard diskless implementation, NFS Diskless.

HP-UX 10.01 offers substantial enhancements in system administration and resource sharing over the base NFS Diskless technology. (See “NFS Diskless”.)

- Multiple file system support. The Journalized File System (referred to as either VxFS or JFS) is an option for the first time.

### Additional 10.01-Specific Changes

- Bundles have been changed. Refer to “Bundles”.
- The support policy for C-Kermit has been clarified. Refer to “Kermit” in Chapter 5.
- There are changes to Curses. Refer to “Curses” in Chapter 5.
- There are changes to Instant Ignition. Refer to “Instant Ignition” in Chapter 5.

## 4-2 Major Changes for HP-UX 10.0 and 10.01

- Variable Length Subnet Masks are supported. Refer to “Variable Length Subnet Masks” in Chapter 6 and “gated” in Chapter 6.
- An interactive interface has been added to Network Tracing and Logging (NETTL). Refer to “NETTRACELOG” in Chapter 6.
- There are new commands: `whois(1)`, `vxdump(1M)`, `vxrestore(1M)`. Refer to “whois” in Chapter 6 for `whois(1)` and “New Commands” for the other commands.
- There are changed commands: `man(1)`, `uucp(1)`, `cu(1)`, `catman(1M)`, and `telnet(1)`. Refer to “Changed Commands in Section 1” in Chapter 7, “Changed Commands in Section 1M” in Chapter 7, and “telnet” in Chapter 6.
- SCSI Pass-Through for Series 800 NIO Bus. Refer to “SCSI Pass-Through for Series 800 NIO Bus” in Chapter 5.

---

## Understanding the Changes

The following topics explain each of the major changes and what they mean to your system and your job as system administrator:

- The HP-UX 10.0 File System Layout
- NFS Diskless
- I/O Convergence
- Kernel Convergence
- Software Distributor

You should read these topics in their entirety, well before you upgrade to 10.01.

Each topic:

- Outlines the change.
- Points you to further information if you need it.
- Explains how to prepare your system for the change, if necessary.

The remaining chapters (4-6) provide details of other changes to HP-UX products, subsystems, commands and libraries. Use the topic lists at the

beginning of each chapter to pick out the topics likely to be of particular concern to you or your users.

---

## Support Information

HP-UX 10.01 supports all computers supported on 9.0 and later releases, with the following exceptions:

- Series 300 and 400 workstations.

No Series 300 or 400 computers are supported on 10.0 or 10.01.

- The following Series 600 and 800 computers, which were also not supported on 10.0:
  - S635SV
  - S645SV
  - S808
  - S815
  - Any Series 800 with an 8-MB memory card in the first slot

Support for the following computers was added at HP-UX 10.0:

- Series 800 Model K.
- Series 700 Model J.

Support for the LaserJet 4MV and ColorLaserJet printers for Series 700 and Series 800 computers has been added at HP-UX 10.01.

HP-UX 10.01 *does not* support the following:

- Upgrade of a 9.x “DUX” cluster.
- Upgrade of Series 700 Model J from 9.x to 10.01.
- C-Kermit 5A(190), which is supported by Columbia University. See “Kermit” in Chapter 5 for details.

### 4-4 Major Changes for HP-UX 10.0 and 10.01



---

## New Commands

This section lists commands that are new to 10.0 or 10.01 and are not discussed elsewhere in this document. See the manpages for details.

New at 10.0:

- `dmpxlt(1)`
- `fstyp(1M)`
- `genxlt(1)`
- `groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`
- `listusers(1)`
- `logins(1M)`
- `mk_kernel(1M)`
- `model(1)`
- `mountall(1M)`, `umountall(1M)`
- `pwconv(1M)`
- `setuname(1M)`
- `sccs(1)`
- `setup(1M)`
- `useradd(1M)`, `userdel(1M)`, `usermod(1M)`

New at 10.01:

- `vxdump(1M)`
- `vxrestore(1M)`
- `whois(1)`

### **dmpxlt(1)**

Converts the compiled version of the `iconv` tables to ASCII text that can be edited and used as input to `genxlt`. Supports single and multi-byte character codesets.

See also `genxlt(1)`.

## **fstyp(1M)**

Determines the file system type in a Multiple File System (MFS) environment.

## **genxlt(1)**

Generates a compiled version of the `iconv` table suitable for use by `iconv(1)` and `iconv(3C)`. Supports single and multi-byte character codesets.

See also `dmpxlt(1)`.

## **groupadd(1M), groupdel(1M), groupmod(1M)**

These commands manage groups:

<code>groupadd</code>	Adds a group to the system.
<code>groupdel</code>	Deletes a group from the system.
<code>groupmod</code>	Modifies an existing group.

You can also perform these functions via SAM.

## **listusers(1)**

Displays information about user logins.

## **logins(1M)**

Displays information about user logins.

## **mk\_kernel(1M)**

Builds a bootable HP-UX kernel, usually `/stand/build/vmunix_test`, and optionally moves it to `/stand/vmunix`. HP recommends you use `mk_kernel` or SAM (the System Administration Manager) to rebuild the kernel on all 10.x systems.

For more information, see the manpage, “Kernel Convergence” of this document, and the *Configuring HP-UX for Peripherals* and *HP-UX System Administration Tasks* manual.

## **4-6 Major Changes for HP-UX 10.0 and 10.01**

### **model(1)**

This command, which was formerly available on the Series 700, is also available on the Series 800.

### **mountall(1M), umountall(1M)**

**mountall**       Mounts all removable file systems listed in the file system table, normally `/etc/fstab`.

**umountall**      Unmounts all removable file systems listed in the file system table.

Both commands comply with SVR4 and support multiple file systems.

### **pwconv(1M)**

Creates or updates the commercial security database from `/etc/passwd`.

- If commercial security is installed, `pwconv` updates the database.
- If commercial security is not installed, `pwconv` removes passwords from `/etc/passwd` and creates the database.

### **setuname(1M)**

Changes system name, node name, or both, either temporarily or permanently (surviving reboot).

### **setup(1M)**

Performs system setup tasks. User interface similar to SAM. Performs these tasks:

- Sets the date and time.
- Sets the timezone.
- Sets the nodename.
- Assigns passwords.
- Adds new users.

### **useradd(1M), userdel(1M), usermod(1M)**

These commands provide a command-line method for adding and deleting users and modifying their accounts.

- useradd**        Adds new users or set defaults for them.
- userdel**        Deletes users, and optionally their home directories.
- usermod**        Modifies the characteristics of an existing user.

### **vxdump(1M)**

Copies to magnetic tape all files in the VxFS file system that have been changed after a certain date. This information is derived from the `/var/adm/dumpdates` and `/etc/fstab` files.

### **vxrestore(1M)**

Reads tapes previously dumped by the `vxdump(1M)` command.

### **whois(1)**

`whois` is similar to a telephone book lookup utility. See “whois” in Chapter 5 for more information.

---

## **The HP-UX 10.0 File System Layout**

A new, completely different file system layout was introduced at HP-UX 10.0. It is modeled after the AT&T SVR4 and OSF/1 file systems and is implemented on both Series 700 and Series 800 computers. The new layout provides benefits such as the separation of operating system software from application software, a foundation for diskless and client/server file-sharing models, and consistency with file system layouts used by many other computer companies.

- You will notice fewer differences between the layouts of other vendors' file systems and that of HP-UX 10.0.
- The file system layout is logical. Files are organized into categories such as static versus dynamic, executable versus configuration data, and so forth.
- To support a client/server file-sharing model, operating system software is kept in separate directories from application software.
- Files that are meant to be shared by different hosts are kept in separate directories from host-specific files.

## Major Changes

Two main areas have changed because of the new file system model:

- file and directory locations
- system startup and shutdown control

### File and Directory Locations

File and directory locations were completely changed at HP-UX 10.0. **Static files**—such as binaries, libraries, documentation, and headers—are separate from **dynamic files**—such as logs, temporary files, and configuration data. Static files, which are also called **shared files**, are grouped together because they usually do not change during system execution. Dynamic files, which are also called **private files**, are grouped together because of their tendency to change size and content during system execution.

A further division is made between the operating system and applications. The static portion of the operating system and applications reside under separate directories. In pre-10.0 releases, applications are co-resident with the operating system, typically under the `/usr` directory. At 10.0, HP applications were moved to a separate directory: `/opt`.

The following table lists most of the HP-UX 10.0 directories and maps them to the equivalent 9.x directories. The 10.0 directories are explained in more detail in the *HP-UX 10.0 File System Layout White Paper*.

**Table 4-1. 9.x to 10.0 Directory Map**

9.x Directory	10.0 Directory	Description/Comments	Private/Shared
/dev	/dev	Device files for local devices	Private
/etc	/etc	Machine-specific configuration and administration databases; no executables invoked by users	Private
N/A	/etc/opt/<application>	Application-specific configuration files	Private
N/A	/etc/rc.config.d	Startup configuration files	Private
N/A	/export	Default root of exported file systems	Server directory
N/A	/export/private_roots	For host-specific files	Server directory
N/A	/export/shared_roots	For shared OS and applications	Server directory
/users	/home	Default for user directories	Private
/users/<username>	/home/<username>	User home directory	Private directory or local mount-point
/lost+found	/lost+found	Storage directory for <b>fsck</b>	Private
/mnt	/mnt	Mounting point for local file systems	Private
/net	/net	Mounting point for remote file systems	Private

**Table 4-1. 9.x to 10.0 Directory Map (continued)**

9.x Directory	10.0 Directory	Description/Comments	Private/Shared
N/A	/opt	Root for optional applications	Private
/usr/<application>	/opt/<application>	Application executables, libraries, and support files	Shared
N/A	/sbin	Essential system commands (those needed to boot system and mount file systems)	Shared
N/A	/sbin/init.d	Startup and shutdown scripts	Shared
N/A	/sbin/rc#.d	Startup and shutdown link files for script sequencing	Shared
N/A	/stand	Standalone machine-dependent binaries and kernel configs	Private
/tmp	/tmp	System-generated temporary files	Private
/usr	/usr	Mount point for sharable user commands, libraries, and documentation	Shared
/usr/bin and /bin	/usr/bin	Operating system user commands	Shared
N/A	/usr/ccs	Unbundled development package	Shared
N/A	/usr/ccs/bin	Development binaries	Shared
N/A	/usr/ccs/lib	Development libraries	Shared

**4-12 Major Changes for HP-UX 10.0 and 10.01**



**Table 4-1. 9.x to 10.0 Directory Map (continued)**

9.x Directory	10.0 Directory	Description/Comments	Private/Shared
/etc/conf	/usr/conf	Kernel configuration	Shared
/usr/contrib	/usr/contrib	Contributed software	Shared
/usr/include	/usr/include	Header files	Shared
N/A	/usr/lbin	Backends to other commands	Shared
/usr/lib and /lib	/usr/lib	Object code and object code libraries	Shared
/usr/local	/usr/local	User-contributed software	Shared
/etc/newconfig	/usr/newconfig	Default operating system configuration data files	Shared
N/A	/usr/old	Obsolete files	Shared
N/A	/usr/sbin	System administration commands	Shared
N/A	/usr/share	Architecture-independent sharable files	Shared
/usr/lib/spell	/usr/share/dict	Dictionaries for <b>spell</b> and <b>ispell</b>	Shared
N/A	/usr/share/lib	Miscellaneous sharable files	Shared
/usr/man	/usr/share/man	Operating system manpages	Shared
N/A	/var	Holds files created at runtime, such as log files and temporary files	Private
/usr/adm	/var/adm	Common administrative files and log files	Private

**Table 4-1. 9.x to 10.0 Directory Map (continued)**

9.x Directory	10.0 Directory	Description/Comments	Private/Shared
/tmp/syscore	/var/adm/crash	Kernel crash dumps	Private
/usr/lib/cron	/var/adm/cron	Cron queueing	Private
N/A	/var/adm/sw	SD directory	Private
N/A	/var/adm/sw/patch	Software patch storage directory	Private
N/A	/var/adm/syslog	Files generated by <b>syslog</b>	Private
/usr/mail	/var/mail	Incoming mail	Private
/usr/news	/var/news	News	Private
N/A	/var/opt/<application>	Application-specific temporary or data files	Private
/usr/preserve	/var/preserve	Preserved editor files	Private
N/A	/var/run	PID files	Private
/usr/spool	/var/spool	Spooled files	Private
/usr/spool/cron	/var/spool/cron	Crontabs and <b>at</b> jobs	Private
/usr/spool/locks	/var/spool/locks	UUCP Lock files	Private
/usr/spool/lp	/var/spool/lp	Printer spooling	Private
/usr/spool/mqueue	/var/spool/mqueue	Outgoing mail	Private
N/A	/var/spool/sw	Default location for SD depot	Private
/usr/spool/uucp	/var/spool/uucp	UUCP spool directory	Private
/usr/spool/uucppublic	/var/spool/uucppublic	Incoming UUCP files	Private
/usr/tmp	/var/tmp	Application-generated temporary files	Private
/usr/spool/uucp	/var/uucp	UUCP admin. files	Private

**4-14 Major Changes for HP-UX 10.0 and 10.01**

## System Startup and Shutdown

The system startup and shutdown control mechanism was also completely new for 10.0. In pre-10.0 releases, `/etc` contains large `rc` scripts that are executed during system boot. System configuration data was embedded in the `rc` scripts along with calls to commands. The mingling of system-specific data and commands made updating the scripts very difficult. This has been replaced by a model that separates the script execution and script variables into different locations. You can set and modify the behavior of the system by changing script variables. The execution scripts, which read the variables and start/stop subsystems, are not user-modifiable. All control is supplied through the script variables.

Execution scripts perform actions on subsystems, such as starting the `cron` and `sendmail` daemons or stopping the NFS daemons. Applications developers may control the run-level at which these subsystem functions are initiated.

The `/etc/shutdown.d` directory is not supported. All scripts in this directory must be redesigned to fit the new startup/shutdown model. See the *HP-UX 10.0 File System Layout White Paper* and the `shutdown(1M)` manpage for more information.

## How This Affects Your System

The biggest impact of the new file and directory locations is on embedded pathnames in program headers and sources, scripts, libraries, object code, binaries, build and test environments, documentation, and path environment variables. For more information, see the *Upgrading from HP-UX 9.x to 10.x* manual.

The 10.0 startup and shutdown model requires you to modify configuration variables that are separate from the execution scripts. The only form of control is through the variables because the execution scripts are not modifiable and will be written over in subsequent releases of HP-UX. If you have modified `rc` scripts to include customized functionality, you need to move the script information to comply with the new startup/shutdown model. For information on the new shutdown and startup model, see the *HP-UX 10.0 File System Layout White Paper* and the *HP-UX System Administration Tasks* manual. If you have scripts in `/etc/shutdown.d`, you will need to migrate them to the

new method. For instructions, see the section on startup and shutdown scripts in Chapter 7 of the *Upgrading from HP-UX 9.x to 10.x* manual.

### **Using 9.x Pathnames for Files in /etc**

You can safely use 9.x pathnames in most circumstances on a 10.x system because **transition links** link the 9.x pathnames to the 10.x pathnames. But if you are in single-user mode, this may not work.

The names of program files and scripts that were in `/etc` in 9.x are in most cases links to corresponding files in `/usr/sbin` in 10.x; for example `/etc/reboot` is a link to `/usr/sbin`. But in single-user mode, `/usr` is unmounted. In this case, you need to use the version of the program that is in `/sbin`.

For example, if you are running in single-user mode and enter the command,

```
/etc/reboot
```

the system will respond:

```
/etc/reboot: not found
```

The command you need in this case is:

```
/sbin/reboot
```

### **Run Levels**

In 10.01, run levels are generally higher by one to perform the same function; for example, initiating run level 4 starts VUE. For more information, see Chapter 1 of the *HP-UX System Administration Tasks* manual.

### **Interoperating with 9.x Systems**

HP has provided tools to make it easier to operate 9.x systems with 10.0 and 10.01 systems.

In addition, files such as `/etc/profile` supplied on 10.01 systems contain code that checks to see if these scripts are running on a 9.x or a 10.x system, and takes appropriate action (setting the `PATH` variable differently, for example).

For more information, see the *HP-UX 9x/10.x Interoperability Guide* supplied with the “HP-UX Upgrade Tools for 9.x to 10.x” package which includes the 9.x-to-10.01 upgrade tools.

### **What Will Happen During the Upgrade to 10.01**

To ease the upgrade to HP-UX 10.x, HP has supplied transition links (essentially symbolic links) that allow applications to reference pre-10.0 pathnames. Although the links provide a vital component to the 10.x upgrade, they do not by themselves guarantee full forward compatibility.

### **Preparing for the Upgrade**

A 10.x system will look different in many ways from a 9.x system; you will need to manage the transition carefully. For example, you may need to modify pathnames and device file names in code and scripts, and if your system is short of disk space you may need to provide more.

Because HP-UX 10.x introduces many changes from previous releases of HP-UX, a number of tools are provided to help you upgrade your system from 9.x releases to 10.x. These tools analyze your 9.x system, warn you of potential problems, and do routine conversion tasks such as converting pathnames to their 10.x equivalents.

Refer to the *Upgrading from HP-UX 9.x to 10.x* manual for specific pre-upgrade tasks and information on the tools that are provided.

---

**Note**            If you are already running 10.x releases, you *do not* need to run any pre-upgrade tools.

---

### **Administering the 10.0 File System**

Refer to the *HP-UX 10.0 File System Layout White Paper* and to the *HP-UX System Administration Tasks* manual for details on file system administration policies and guidelines.

---

## NFS Diskless

As of 10.01, HP has replaced its proprietary technology for “diskless” clusters, commonly known as “DUX” (Distributed HP-UX), with the de facto UNIX standard, NFS Diskless. But HP is retaining the SAM system management tool as the mechanism for cluster management. SAM extends the base feature set of NFS Diskless so as to retain a key strength of “DUX”: easy administration from a single seat. This set of enhancements is called **Single Point Administration**, or **SPA** for short.

## Other Documents

- For information on upgrading a “DUX” cluster to NFS Diskless, see *Upgrading from HP-UX 9.x to 10.x*.
- For an introduction to NFS Diskless concepts, and important background on system administration, see the *NFS Diskless Concepts and Administration White Paper*, which is on your 10.x system in PostScript format in `/usr/share/doc/NFSD_Concepts_Admin.ps`. This file is also available in the “HP-UX Upgrade Tools for 9.x to 10.x” package.
- For information on optimizing NFS client/server configurations, refer to the *NFS Client/Server Configuration, Topology, and Performance Tuning Guide White Paper*, which is on your 10.x system in PostScript format in `/user/share/doc/NFS_Client_Server.ps`.
- For information on setting up and running a new cluster see Chapter 11, “Setting Up and Administering an HP-UX NFS Cluster,” in the 10.01 version of the *HP-UX System Administration Tasks* manual, and the online help in SAM.

## What Is NFS Diskless?

NFS Diskless is the industry standard technology for sharing file-system and other hardware and software resources amongst a group of computers connected by one or more LANs (Local Area Networks).

NFS Diskless is like HP’s proprietary “DUX” technology (obsolete as of 10.0) in that it:

- Allows clients to boot remotely from a kernel stored on a remote server

## 4-18 Major Changes for HP-UX 10.0 and 10.01

- Allows the sharing of the root file system
- Allows the sharing of other resources such as:
  - Swap
  - Non-root file systems
  - Spooled printers

But the underlying mechanisms are different:

Feature =====	"DUX" Mechanism =====	NFS Diskless Mechanism =====
Shared file system	Proprietary global mount	NFS mounts
Private kernel, device files, etc.	Context-Dependent Files (CDFs)	HP-UX 10.0 File System Layout
Remote boot	RMP	BOOTP/TFTP
Remote swap	Device swap	File-system swap
Client/server communication	Link-level LAN address	NFS/TCP-IP

For more details, see ““DUX” versus NFS Diskless: Summary of Key Differences”.

### Reasons for the Change

- Provides industry-standard file-sharing and remote boot technology
- Provides basis for future sharing of heterogeneous operating systems and releases (but HP only, current release only, at 10.x)
- Allows much greater flexibility in configuring and distributing hardware

### New Features and Capabilities

- A server can support clients on multiple LANs
- A client can boot from a server on a different subnet
- Any system running HP-UX 10.01 or later can swap to any NFS-mounted file system

- Clients boot using standard BOOTP/TFTP protocols  
(Existing Series 700 clients with RMP in boot ROM will still boot correctly)
- Clients can mount local CD-ROM file systems
- Series 800 systems can be servers

### **New Parameter**

The kernel parameter `page_text_to_local` controls whether, and where, executable code is swapped. How you set this parameter can affect the performance of an NFS Diskless client:

- If `page_text_to_local` is off (0), executable code pages will not be swapped out, but must be re-read from disk if they have been cleared out of memory.

This is the default.

- If `page-text-to-local` is on (1), executable code pages *will* be swapped out, to the local disk, when they are cleared out of memory.

From this you can see that a client that swaps to its local disk, but executes applications that are not on the local disk, will perform much better if you turn `page_text_to_local` on (set it to 1).

### **Old (“DUX”) Features No Longer Supported**

- Context-Dependent Files (CDFs)

NFS Diskless uses the mechanisms provided in the 10.0 file system layout to separate private from shared files (see “The HP-UX 10.0 File System Layout”.)

Under 10.x, any CDFs left over from your 9.x system will appear as regular directories that have the SUID bit set.

- Support for Series 300 and 400 systems as clients and servers

HP-UX 10.x does not support Series 300 and 400 computers. See the *Upgrading from HP-UX 9.x to 10.x* manual for more information.

- Automatic cluster-wide sharing of mounted file systems

## **4-20 Major Changes for HP-UX 10.0 and 10.01**



File systems must be shared explicitly (for example via NFS), but SAM-SPA provides global management services.

- Cluster-wide shutdown
- POSIX-compliant file system
- Remote device swap

NFS diskless uses file-system swap.

- LAN break detection

But clients will not panic if the LAN goes down; they will simply wait for communication to be re-established.

- Support for configuring and reviewing Access Control Lists on (ACLs) on NFS clients

Commands such as `chacl(1)` and `setacl(1)` will fail on an NFS-mounted (imported) file system. But you can still set up ACLs on the NFS server and they will be enforced on the NFS clients.

- Distributed named pipes

“DUX” allowed one client to read a named pipe written by another; NFS Diskless does not.

- Global PIDs (Process Identifiers)

PIDs are managed on an NFS Diskless node just as they are on a standalone machine.

- Automatic clock synchronization.

You must synchronize the client’s clock with the cluster time when you add the client via SAM.

- Certain tunable parameters. (See “Details of Changes for NFS Diskless”.)

For more information:

- If you are planning to upgrade a 9.x “DUX” cluster to 10.x, see the *Upgrading from HP-UX 9.x to 10.x* manual for details.
- For more detailed information on changes to commands, files and kernel parameters, see “Details of Changes for NFS Diskless”.
- For a detailed comparison between “DUX” and NFS Diskless, see ““DUX” versus NFS Diskless: Summary of Key Differences”.
- For an overview of the V.4 file system layout, see “The HP-UX 10.0 File System Layout”.

### “DUX” versus NFS Diskless: Summary of Key Differences

The following table summarizes the differences between “DUX” and NFS Diskless clusters, and shows how the SAM-SPA system management tool both extends base NFS functionality and makes it easier to manage.

“DUX” Feature	NFSD equivalent	SAM-SPA enhancement	9.x-10.x change
Series 700 can be clients or servers	Series 700 can be clients or servers		
Series 300/400 systems can be clients	Series 300/400 not supported		Series 300/400 cannot upgrade to 10.x and cannot boot remotely from 10.xut can import and export file systems.
Series 800 systems not supported at 9.0	Series 800 supported as servers only		Series 800 can be cluster server, can import and export file systems and swap remotely, but cannot boot from remote server

"DUX" Feature	NFS equivalent	SAM-SPA enhancement	9.x-10.x change
Proprietary remote boot (RMP)	bootp/tftp		Supported on S700 only (S800s can be servers).  rbootd(1M) modified so existing S700s can boot via NFS.  Clients can boot across subnet.  Server can support clients on more than one LAN card.
File sharing via global mount	File sharing via NFS	Global file system management services  Temporary mounts (not entered in /etc/fstab) not supported by this service.	HP-UX now uses industry-standard NFS mechanism but allows global management via SAM-SPA
Shared/private file separation via CDFs (context-dependent files)	V.4 file system layout separates private from shared directories.		CDFs no longer supported

"DUX" Feature	NFSD equivalent	SAM-SPA enhancement	9.x-10.x change
Remote or local device or dynamic swap	Remote dynamic swap or local device or dynamic swap		Remote device swap not supported.  Any node can swap dynamically to any remote system (not just to those with common cluster server).  Server does not have to swap to own disk space.  Swap files not freed on reboot.
CDROM file system (CDFS) support for mounts on server only	Mounts supported on any node		CDROM volumes can be mounted on any 10.x system.

#### 4-24 Major Changes for HP-UX 10.0 and 10.01

"DUX" Feature	NFSD equivalent	SAM-SPA enhancement	9.x-10.x change
POSIX-compliant file system	None		The NFS diskless file system is not POSIX-compliant.
Shared spooling and printing	None	Global printer management services permit sharing	
Cluster-wide backup	(Can NFS-mount all file systems to one server)	Cluster-wide backup service backs up all private roots and all shared file systems.	Service runs on cluster server only, though shared file systems can be backed up from any node.
Shared mail	None	Mail service supports both cluster-wide and private mail management.	New "private" mail management service (for a single node) gives greater flexibility than before.
"DUX" Feature	NFSD equivalent	SAM-SPA enhancement	9.x-10.x change
Cluster-wide user and group id's	(NIS)	Support for global user and group id's and home directories. "Private" id's can also be configured.	"Private" option is new. Global id's depend on symbolic links to group and password files; be careful not to unlink.
Cluster-wide time synchronization	None	Support for synchronizing cluster time via NTP	Clocks do not synchronize automatically as before; must be configured (via SAM) when client is configured.
Distributed named pipes	None		Distributed named pipes not supported.

"DUX" Feature	NFSD equivalent	SAM-SPA enhancement	9.x-10.x change
=====			
Global Process Identifiers (PIDs)	None		Each node will manage its own PIDs
Shared /tmp	/tmp can be NFS mounted		Must NFS mount /tmp explicitly if you need to share it.
Shared Access Control Lists (ACLs)	None		ACLs cannot be <i>configured</i> on clients, but will <i>work</i> on clients when configured on server.
Menu-driven cluster config.	None	Similar menus	
		Remote Task Monitor	New SAM feature for clusters.

## Details of Changes for NFS Diskless

### New Commands

The commands `dcnodes(1M)` and `dcnodesd(1M)` replace `cnodes`. `dcnodes(3X)` is the corresponding library routine. Refer to the `dcnodes(1)` manpage for more information.

`swcluster(1M)` manages installing and removing software in NFS Diskless clusters.

### Changes to Commands and Files

The following commands have new options to support NFS Diskless:

Command	Option	Function
=====	=====	=====
<code>swapon(1M)</code>	<code>-t</code>	Allows choice of type of swap entry

## 4-26 Major Changes for HP-UX 10.0 and 10.01

(in /etc/fstab--formerly /etc/checklist)  
to be processed

mount(1M)        -Q        Causes mount to ignore file systems  
already mounted

### Commands, Parameters, Options No Longer Supported

The following commands were specific to “DUX” and are no longer supported:

ccck(1M)  
cdf(4)  
cfuser(1M)  
cluster(1M)  
cnodes (1)  
context(5)  
cps(1)  
csp(1M)  
cwall(1M)  
getcontext(1)  
makecdf(1M)  
showcdf(1M)

The following commands had “DUX”-specific options that are no longer supported:

Command	Options
=====	=====
bdf(1M)	-L
chmod(1)	-H
fbackup(1M)	-H
find(1)	-hidden -type H
ftio(1)	-H
last(1M)	-c
ls(1)	-H
mount(1M)	-L
pax(1)	-H
pwd(1)	-H
reboot(1M)	-l -b
sync(1M)	-l

syncer(1M)	-l
tar(1)	-H
umount(1M)	-L
users(1)	-c
who(1)	-c

The following library and system calls are disabled or unsupported:

```

cnodes(2)
cnodeid(2)
getcontext(2)
diskless_stats(2)
endccent(3c)
fgetccent(3c)
ftwh(3c)
getccent(3)
getccid(3c)
getccnam(3c)
getcdf(3c)
getcontext(2)
getcwd(3c)
hidecdf(3c)
lsync(2)
mkrnod(2)
nftwh(3c)
setccent(3c)
setcontext(2)

```

`cnodes` and `cnodeid` return zero if called on 10.01, and `getcontext`, `setcontext`, `diskless_stats`, `lsync` and `mkmod` return the `EOPNOTSUPP` error. The remaining calls in the above list are not recognized on 10.01.

The following kernel parameters were specific to “DUX” and are no longer supported:

```

num_cnodes
serving_array_size
server_node
using_array_size
dksless_fsbufsn
gcsp

```

#### 4-28 Major Changes for HP-UX 10.0 and 10.01



```
rdu
server_node
check_alive_period
retry_alive_period
selftest_period
reboot_option
```

## Upgrading a Cluster to 10.01

HP has provided tools to upgrade a 9.x “DUX” cluster to 10.01 NFS Diskless. You can then upgrade the NFS Diskless cluster. To upgrade from “DUX”, you need the package “HP-UX Upgrade Tools for 9.x to 10.x” and the manual *Upgrading from HP-UX 9.x to 10.x*, which is part of the package.

## A Useful Script

The `nfsstat` command displays statistical information about the NFS and Remote Procedure Call (RPC) interfaces to the kernel. The *NFS Client/Server Configuration, Topology, and Performance Tuning Guide White Paper* (on your 10.01 system in `/usr/share/doc/NFS_Client_Server.ps`) provides a script for collecting these statistics. Here is the source for that script.

```
#!/usr/bin/ksh
#
#This is a sample script that runs the nfsstat command at given
#intervals and normalizes the output.
#
# Get a series of nfsstat reports and normalize over time.
# Results are reported "per second", whatever the window used.
#
# Invoked as: Program-name interval nfsstat_flags runtime
# All time units are in seconds.
#
# Output:
# The output is similar to the nfsstat format appropriate to the option
# requested with the addition of a % value which is the percent that
# particular field is of the number of calls.

export PATH="/usr/bin/:$PATH"
if [ "$1" = "-f" ] ; then
    fileinput=1
    shift
    echo using files $*
```

```

else
    if [ "$1" != "" ] ; then
        window=$1
    else
        window=${WINDOW:-60}
    fi
    echo Window=$window
    ttime=$3
    if [ "$ttime" = "" ] ; then
        ttime=9999999
    fi
    starttime=$SECONDS
fi
( if [ "$fileinput" = "" ] ; then
    while [ "$ttime" -gt $SECONDS ] ; do
        let w2=ttime-starttime
        if [ "$w2" -le "$window" ] ; then
            echo Last window = $w2 >&2
            window=$w2
        fi
        echo "elapsedtime: $et"
        /usr/bin/nfsstat $2
        echo '--'
        sleep $window
        endtime=$SECONDS
        let et=endtime-starttime
        starttime=$SECONDS
    done
else
    echo "elapsedtime: 600" # Assumes data taken over 10 minute interval
    cat $*
fi
) | awk '
BEGIN          { firsttime=1 }
$1 == "--"    { firsttime = 0; next}
$1 == "elapsedtime:" {et=$2;print "";}
              if (firsttime == 1 ) { printf ("Started at ")}
              system("date +%D\ %X"); next}
/^$/         {if (firsttime != 1) { print "" };next}
/^Client |^Server / {class=$0
                  if (firsttime != 1) { print class }
                  next}
/^[a-zA-Z]/  {labelstr=$0
              for (i=1; i<= NF; i++) {
                  label[i]=$i
                  if (label[i] == "calls" ) {
                      totcalls=deltaval
                  }
                  if (firsttime!=1) {printf("%8s/s ",label[i])}

```

#### 4-30 Major Changes for HP-UX 10.0 and 10.01

```

} if (firsttime != 1) { print "" }
numlabels=NF
getline
datastr=$0
for (i=1;i<=numlabels;i++) {
    if (NF == 2*numlabels) {
        datapos=i*2-1
    }
    else {
        datapos=i
    }
    val=$datapos
    if (firsttime != 1) {
        deltaval=val-lastval[class label[i]]
        if (label[i] == "calls" ) {
            totcalls=deltaval
            if (totcalls == 0 )
                { totcalls=1 }
        }
        printf ("%6.2f %2d%% ", deltaval/et,\
            int(100*deltaval/totcalls+0.5))
    }
    lastval[class label[i]]=val
}
if (firsttime != 1) { print "" }
next
}
{print "error on line: " $0}

```

---

## I/O Convergence

Before 10.0, HP-UX provided different I/O subsystems on different hardware platforms. This meant, for example, that the Series 700 did not support Series 800 drivers. HP-UX 10.0 introduced a single, unified I/O subsystem that allows both Series 700 and Series 800 I/O drivers to run in the same kernel.

### Major Changes for I/O Convergence

The major features of I/O convergence are:

- A single set of kernel configuration utilities.

SAM will do kernel builds via `config(1M)`. You can also use `config` yourself, or, preferably, `mk_kernel(1M)`. SAM, `mk_kernel` and `config` are supported on both Series 700 and 800. `uxgen` is not supported on 10.x. See “Kernel Convergence” for more information.

- A single convention for creating and naming device files.

The convention comprises:

- A single set of major numbers, and a single method of assigning minor numbers, across both platforms. See “Major and Minor Numbers”.
- A new convention for device file names, based on the hardware path. See “Device File Naming Convention”.

- A single scheme for disk layout.

See “Disk Management”.

- Some changes in driver naming and support.

See “Drivers”.

### Major and Minor Numbers

As of 10.0, HP-UX supports a single set of major numbers, and a single method for assigning minor numbers, across the Series 700 and 800 platforms. This means, for example, that the device file for a SCSI disk will look the same on both kinds of system, in some cases allowing devices to be interchanged freely between Series 700 and 800 systems, and in general making it easier to write drivers and applications for HP systems.

Converging major and minor numbers entails the following specific changes:

- Series 800 Logical Unit (LU) numbers are no longer supported.

They are replaced by instance numbers. See “LU Numbers and Instance Numbers” below.

- Peripherals must be accessed via different major and minor numbers in 10.x from those that would be used on a 9.x system.

This should not cause you problems unless you use `mknod` in code or scripts; run `prepare(update_aid)` to check for instances of `mknod`.

See the *Configuring HP-UX for Peripherals* manual for details of the new numbering scheme.

- Some major numbers are being reserved for HP drivers.

But major numbers that have been reserved for non-HP drivers in the past have not been touched.

- Some major numbers will be assigned dynamically at boot time.

But device files that use these major numbers will keep the same name in 10.x as they have on your current 9.x system.

See “How This Affects Your System”, “I/O Convergence and 9.x Code and Scripts”, and “Avoid mknod” for more information.

### **LU Numbers and Instance Numbers**

LU numbers were assigned to all devices on Series 800 systems at the time of bootup. As of 10.0, this no longer happens. Instead, on both Series 700 and 800 systems, the HP-UX I/O system assigns each device an instance number, which (like LU numbers) does not change when you reboot the system.

Instance numbers of interface cards (that is, the cards to which peripherals such as disks and printers are connected) play an important role in binding the device driver to the device; they are represented in the minor number and, by the 10.x naming convention, in the device file name (see “Device File Naming Convention” later in this section).

### **Interface versus LU.**

- Instance numbers are used (and can be seen in `ioscan`) on all HP-UX 10.x systems, whereas LU numbers were assigned only on Series 800 systems.
- LU numbers on pre-10.0 systems are comparable to device instance numbers on 10.x systems, in that they identify each device type uniquely at bootup.  
But the only instance numbers you as a system administrator need to be concerned with are those assigned to interface cards.
- LU numbers changed at each boot. Instance numbers for interface cards are preserved across bootups, in the `ioconfig` files.

**Interface Classes.** As you can see by running `ioscan(1M)` on a 10.x system, the `INTERFACE` hardware type is subdivided into `classes`. Examples of interface classes are `ext_bus`, `lan`, `tty`, `ps2`, `graphics`, `lantty`, `lanmux`, `audio`, `pc`, and `hil`. The I/O system assigns separate sets of instance numbers to each class of interface; you can see this by looking in the `I` column in the output from `ioscan -f`.

The following are some points to note about interface numbers and classes:

- Class is assigned by interface driver that claims a card, not by the card itself.  
This allows driver developers to introduce their own classes.
- Instance numbers are repeated between classes but are always unique within a class.
- Instance numbers are not necessarily consecutive.
- Instance numbers are assigned in the order cards are bound to their drivers.

### **Guidelines for Working with 10.x Device Files.**

- Always check the output of `ioscan(1M)` before interpreting an instance number.
- Use `insf(1M)`, `mksf(1M)` or SAM whenever possible to create device files; this will protect you from making mistakes with major and minor numbers.
- When creating a device special file, use the instance number of the *interface card to which the device is attached*, not the instance number of the device itself.

For more information, see the section “`insf(1M)`, `lssf(1M)`, `mksf(1M)`, `rmsf(1M)`” in Chapter 7 in this document, the manpages, and the *Configuring HP-UX for Peripherals* manual.

### **How This Affects Your System**

When you upgrade from 9.x, the upgrade software will create new device files for the devices on your system, changing the names and major and minor numbers to conform to the 10.0 conventions. (See the *Upgrading from HP-UX 9.x to 10.x* manual for details and exceptions. See “Device File Naming Convention” for information on the new naming convention.)

The upgrade software will also set symbolic links that link the old device-file names on your system to the new 10.01 device files. But you need to check code and scripts for calls to `mknod`; these will probably create device files that are invalid for 10.01.

### **I/O Convergence and 9.x Code and Scripts**

9.x code and scripts that deal explicitly with I/O devices and drivers will probably need to be modified before you can run them on a 10.x system. When checking for compatibility, remember that:

- Many 9.x device-file names are not valid on 10.x.  
See “Device File Naming Convention”.
- 9.x device files, and 9.x calls to `mknod(1M)`, probably use major or minor numbers that are not valid on 10.x.
- Not all drivers supported before 10.0 are supported on 10.x.

See “Drivers”.

HP has provided tools to help you make these checks. See “Preparing for I/O Convergence”.

Also see “Avoid mknod” for more information.

**Avoid mknod.** When you add a new peripheral to a 10.01 system, you should normally use `insf(1M)` or `mksf(1M)`, or SAM, rather than `mknod`, to create the device file for you. This will keep you from having to make literal reference to unfamiliar major and minor numbers.

See the *Configuring HP-UX for Peripherals* manual for instructions on configuring new devices into your 10.x system, and a full explanation of the new major and minor numbering scheme.

## Device File Naming Convention

In all past releases, device files on the Series 700 have been named by one convention, and device files on the Series 800 by another. For example, a printer connected to a serial interface on a Series 800 would have a device file name such as `/dev/tty6p1`, whereas on a Series 700 the name would be something like `/dev/ptr_rs232a`.

In HP-UX 10.x, device file names on both the Series 700 and 800 have the form:



*prefix* [*id* | *devspec*] [*options*]

where:

*prefix* is `/dev/` or a directory relative to it (`/dev/diag/`, for example)

*id* is an arbitrary number or designator, for example `0m` in the default tape device file `/dev/rmt/0m`. *id* should be used only for system default device files or device files used by pseudo drivers that have no direct relationship to hardware. In all other cases, use *devspec*.

*devspec* indicates the hardware path, in the form,

`c#[t#[d#]]`

where:

**c#** is the instance number of the card to which the device is attached (unique per SPU). The number is base-10 and can be from one to three digits. It is assigned by the I/O system and does not change when you reboot. It is the number in the I column in the output of `ioscan(1M)`. (See “Major and Minor Numbers” earlier in this section.)

**t#** is the address set on the device (for example the HP-IB address, or SCSI “target” number). The number is base-10 and can be one or two digits; the typical range is 0-15.

**d#** is the device unit number (for example, SCSI LUN or HP-IB unit).

*id* or *devspec* (preferred) must form part of the name unless *prefix* is sufficient to specify the device.

*options* are device specific, for example, `_lp` to indicate a printer, or `s#` to indicate a disk section number. (Leaving the `s#` out of a disk device file name implies that the file addresses the whole disk.)

The following section shows some examples.

## Examples

<code>/dev/rdisk/c0t6d0</code>	SCSI disk on interface card 0, SCSI address 6, SCSI LUN 0 (whole disk access indicated by absence of <b>s#</b> ).
<code>/dev/c0t1d0_lp</code>	SCSI printer on interface card 0, SCSI address 1, SCSI LUN 0.
<code>/dev/rmt/c0t2d0m</code>	SCSI tape drive on interface card 0, SCSI address 2, SCSI LUN 0, medium density.

**System Default File Names.** System default file names such as `/dev/lp` and `/dev/rmt/0m` will continue to exist in 10.x; they will be symbolic links to the actual device files named according to the convention just described. For example, `/dev/rmt/0m` might be a link to `/dev/rmt/c0t2d0BEST`.

## How This Affects Your System

When you upgrade your system to 10.01, the upgrade software will modify both the *contents* of most device files (because major and minor numbers are changing—see “Major and Minor Numbers”) and their *names* (in accordance with the new convention described in this section). But the upgrade software will set symbolic links that link the old names to the new device files.

So code and scripts that invoke the old names will continue to work, but you need to check for calls to `mknod`. HP has provided tools to help you do this. See “Preparing for I/O Convergence”.

The upgrade software will also remove device files used by drivers that are no longer supported (see “Drivers”).

**Creating Device Files.** You should normally avoid using `mknod(1M)` to create device files on 10.01. Use `insf(1M)` or `mksf(1M)`, or SAM, instead. This will keep you from having to make literal reference to unfamiliar major and minor numbers.

See the *Configuring HP-UX for Peripherals* manual for instructions on configuring new devices into your 10.x system.

## **Disk Management**

Logical Volume Manager (LVM) is offered on both the Series 700 and the Series 800 as of 10.0, and HP encourages you to adopt this method of disk management. Older methods, such as hard partitions (Series 800) and SDS (Series 700) are being phased out; see “Support Matrix for Disk Management Methods” later in this section for more information.

## New Features and Capabilities of LVM

- Supported on Series 700.

All features of LVM (including root, dump and swap, and the new features in this list) are available on both the Series 700 and the Series 800. For LVM commands that are new on the Series 700, and changed on the Series 800, see “LVM Commands” later in this section.

- Disk striping of a logical volume.

LVM striping replaces SDS (Software Disk Striping). Under LVM you can set the stripe size to 4, 8, 16, 31, or 64 KB. The default is 8 KB. The number of disks you can stripe across must be at least two but cannot exceed the number of disks in your volume group.

- Disk mirroring.

LVM and the MirrorDisk/UX product support all features of DataPair/800. DataPair/800 itself is no longer supported. New features include:

- `lvsplit(1M)` can operate on more than one logical volume.
- A volume group can be activated as read-write on one system and read-only on another.

This allows you to do backups on the read-only system; but note that a single *logical volume* can be accessed from only one system at a time.

For more information, see Chapter 7, “Mirroring Data Using LVM,” in the *HP-UX System Administration Tasks* manual.

- Physical volume links for HP High Availability Disk Arrays.

Allows dual links (or paths) to the same disk, such that if one link to the disk fails, LVM automatically routes the I/O to an alternate link.

- Disk locking (Series 800).

The optional product *MC/ServiceGuard*, available at 10.0 on the Series 800 only, provides a disk locking mechanism within LVM to ensure that disks that are connected to two or more systems are accessed by only one system at a time. For more information, see the *Managing MC/ServiceGuard* manual (B3936-90001).

## 4-40 Major Changes for HP-UX 10.0 and 10.01

## LVM Commands

The following commands were new on the Series 700 at 10.0 and, where noted, changed on the Series 800. Refer to the manpages for more details.

- lvcreate(1M)—revised to support disk striping
- lvchange(1M)
- lvdisplay(1M)—revised to support disk striping
- lvextend(1M)
- lvlnboot(1M)
- lvmerge(1M)—revised to support addition of atomic sync
- lvmmigrate(1M)
- lvreduce(1M)
- lvremove(1M)
- lvrmboot(1M)
- lvsplit(1M)—revised to support addition of atomic sync
- lvsync(1M)
- pvchange(1M)
- pvcreate(1M)
- pvdisplay(1M)
- pvmove(1M)
- vgcfgbackup(1M)
- vgcfgrestore(1M)
- vgchange(1M)—revised to support addition of off-line backup
- vgcreate(1M)
- vgdisplay(1M)—revised to support addition of off-line backup
- vgexport(1M)
- vgextend(1M)
- vgimport(1M)
- vgreduce(1M)
- vgscan(1M)
- vgsync(1M)

In addition, LVM commands have been modified to call `vgcfgbackup` automatically whenever you make a configuration change; see Chapter 3 in the *HP-UX System Administration Tasks* manual for details.

### Products and Features No Longer Supported

#### ■ DataPair/800

This product was no longer supported as of 10.0, but 10.x LVM supports its features on both the Series 700 and Series 800 via the optional product *MirrorDisk/UX*. See “Preparing for I/O Convergence”.

Mirrored striped disks are *not* supported on 10.x.

#### ■ Software Disk Striping (SDS) (Series 700).

This product was replaced by LVM disk striping as of 10.0. New SDS disks cannot be configured into 10.01, but:

- Existing *single* SDS striped disks will continue to be supported by means of a compatibility driver (`cpd`) which will be automatically bound into the new kernel when you upgrade from 9.x if you have disks in this configuration. (This does not apply to SDS on the root disk because that configuration has never been supported.)
- Existing SDS disk *arrays* will be automatically converted for you to LVM volume groups during the upgrade from 9.x.

- Hard partitions.

Hard partitions are supported on 10.x for all disks that supported them as of 9.04. They are not supported for new disks introduced at 10.0 or later, or for disks that did not support them at 9.04.

*For 10.01:*

- `insf(1M)` will not create device files for disk partitions (it will create a single device file for the whole disk) but you can create them yourself, for disks that support them, using `mksf(1M)`.
- The definitions of sections 0 and 2 were swapped as of 10.0; see “Other Changes” later in this section.
- You cannot *install* 10.x to a hard-partitioned root disk (that is, “cold install” as opposed to upgrading); SD will offer you the choice of two types of access to the root disk:
  - Whole disk access; or
  - LVM
- You *can* upgrade from 9.x to 10.x using a hard-partitioned disk that was supported as of 9.04.

### Support Matrix for Disk Management Methods

Logical Volume Manager (LVM) is the disk management scheme HP recommends for 10.0 and later releases running on both the Series 700 and 800. Older methods are being phased out.

The following table shows what level of support, if any, HP-UX 10.x provides for disk management schemes that were supported in 9.x: SDS, hard partitions, whole-disk, and LVM.

Management Scheme	10.x Supports?	Can Convert to LVM?	Can Install 10.x to?
SDS:	yes	yes (arrays)	no
Hard partitions:	yes	yes	no
Whole disk:	yes	yes	yes
LVM:	yes	-	yes

## Other Changes

- The names and contents of your disk device files will change when you upgrade to 10.01. See “Device File Naming Convention” and “Major and Minor Numbers”.
- For hard-partitioned disks (Series 800), the definitions of sections 2 and 0 have been swapped:
  - Section 0 = whole disk access (old section 2)
  - Section 2 = old section 0

This is to create consistency between the Series 700 and the Series 800.

No data will be moved or overwritten. **Transition links** created by the upgrade software will ensure that code accessing the whole disk via section 2 at 9.x will continue to read and write to the whole disk at 10.01, and that disk access to 9.x section 0 is redirected to 10.01 section 2.

## How This Affects Your System

The favored disk-management method for 10.01 is LVM. But if you want to continue as before, and you are not using *DataPair/800* for disk mirroring, you do not need to do anything to prepare your disks for the upgrade to 10.01: your 9.x configuration will either be supported by compatibility drivers or, in the case of SDS disk arrays, automatically converted for you when you upgrade.

If you are using *DataPair/800*, you must migrate to *MirrorDisk/UX* before you upgrade. *MirrorDisk/UX* is an optional product that runs under LVM. For instructions on migrating from *DataPair/800* see Chapter 8, “Managing Logical Volumes,” in the 9.0 Series 800 version of the *HP-UX System Administration Tasks* manual.

If you want to make broad disk-management changes (for example from hard partitions to LVM) the upgrade to 10.01 may or may not be a good time to do it. For information and advice, see “Disk Configurations in 9.x and 10.01”.

See “Disk Configurations in 9.x and 10.01” for more information.

**Disk Configurations in 9.x and 10.01.** The table below shows existing (9.x) configurations and possible 10.01 configurations. The “Action Needed” column shows what, if anything, you need to do to get from one to the other.

## 4-44 Major Changes for HP-UX 10.0 and 10.01



Series 700:  
 =====

----- 9.x -----		----- 10.x-----		What HP-UX Does	Action Needed
Root Disk	Data Disks	Root Disk	Data Disks	=====	=====
Single disk	Single disks	Single disk	Single disks	Same as 9.x	None
Single disk	SDS array	Single disk	LVM striped volumes	Converts SDS array to LVM during upgrade via sdstolvm See Note 1.	None
Single disk	SDS single disk	Single disk	Single disk	Supports SDS disk via compatibility driver	None
Single disk	SDS single disk	Single disk	LVM striped volumes	Supports disk via LVM	Convert disk to LVM. See Note 2.
Single disk	Any	LVM disk	Any	Supports disk via LVM	See Note 3.
Single disk	Single disks	Single disk	LVM disks	Supports disks via LVM	Convert disks to LVM. See Note 2.

Series 800:  
 =====

----- 9.x -----		----- 10.01-----		Action	
Root Disk	Data Disks	Root Disk	Data Disks	What HP-UX Does	Needed
=====	=====	=====	=====	=====	=====
Partitions	Partitions	Partitions	Partitions	Supports disks via compatibility driver	None
Partitions	LVM	Partitions	LVM	Supports disks via compatibility driver, LVM	None
LVM	LVM	LVM	LVM	Supports disks via compatibility driver, LVM	None
Partitions	Partitions	Partitions	LVM	Supports disks via compatibility driver, LVM	Convert disks to LVM. See Note 2.
LVM	Any	LVM	LVM striped	Supports disks via LVM	Convert disks to LVM. See Note 2.
Partitions	Any	LVM	Any	Supports disk via LVM	See Note 3.

### Configurations Not Supported in 10.x

- SDS arrays.

These will be converted to LVM during the upgrade. See Note 1.

- SDS root disk.

This configuration has never been supported.

#### Notes

1. SDS disk arrays.

During the upgrade to 10.01, the `sdstolvm` utility runs automatically and converts SDS disk arrays to an LVM volume group, removing SDS device files and updating `/etc/fstab` (the 10.x version of `/etc/checklist`).

These disks will be managed by the Logical Volume Manager (LVM) under 10.01. See Chapter 3, “Managing Disks Using the Logical Volume Manager (LVM),” in the *HP-UX System Administration Tasks* manual.

#### 2. Data disks (other than SDS disk arrays).

You do not have to convert these disks to LVM in order to upgrade to 10.x. If you want to convert your disks, do the following:

- a. Upgrade to 10.x.
- b. Back up the data on the disks.
- c. Configure the disks as LVM (or LVM striped) disks.

Use SAM if you have not done this before. For more information, see Chapter 3, “Managing Disks Using the Logical Volume Manager (LVM),” in the *HP-UX System Administration Tasks* manual.

#### d. Recover the data.

#### 3. Root (system) disk.

You do not have to convert your root disk to LVM in order to upgrade to 10.01, and you cannot do it during the upgrade.

If you want to convert the root disk, you must do it either before or after upgrading to 10.01 on a Series 800. See the *Upgrading from HP-UX 9.x to 10.x* manual for instructions on converting the root disk while running 9.x. On a Series 700, you must upgrade to 10.01 before you can use LVM. See the *HP-UX System Administration Tasks* manual for directions for converting once you are on 10.01.

## Drivers

HP-UX 10.x allows drivers written for both the Series 700 and the Series 800 to run in the same kernel. This implies some modifications for existing drivers.

## Checklist for Driver Developers

Developers of 9.x in-house and third-party drivers should modify these drivers as follows:

- Take account of the new converged minor number scheme.
- Include new header information and an install entry point.
- Remove hard-coded bus-dependent service calls.
- Remove calls to services that directly access the I/O tree or manipulate manager indices.
- Modify the way the driver obtains and stores device instance numbers (see “LU Numbers and Instance Numbers” earlier in this chapter).
- Associate the driver header with the `isc` entry for the hardware.
- Take account of the U2 I/O adapter and cache coherence on the Series 800 Model K and the Series 700 Model J if the driver needs to run on those platforms.

For detailed guidance, Series 700 driver developers should refer to the 10.01 version of the *Driver Development Guide*. HP does not publish guidelines for Series 800 drivers.

## Unmodified Drivers

Unmodified 9.x drivers will continue to run on 10.01 if they

- are EISA, non-SCSI drivers; and
- were written in accordance with HP’s guidelines in the then-current version of the *Driver Development Guide* and
- do not use a major number that is pre-empted by an HP driver in 10.01 (see “Major and Minor Numbers” earlier in this chapter).

But developers will need to do some work to take account of the restructuring of the information that is stored in `/etc/master` in 9.x. A new scheme is in effect as of 10.0; see “Kernel Convergence” later in this chapter.

## New Drivers

HP has added the following drivers:

System	New Driver	Description
=====		
700/800	cpd	Compatibility pseudo-driver (for compatibility with 9.x SDS and hard partitions).
700/800	kepd	Kernel entry point driver.

For more information on compatibility between 9.x and 10.01 disk management schemes, see “Disk Management” earlier in this chapter.

## Driver Name Changes

System	Old Name	New Name	Description
=====			
700	parallel	centif	Centronics parallel port driver.
700/800	autoch	ssrfc	MO surface driver.
700	scsi	sdisk	SCSI disc driver.
700	autox	schgr	MO autochanger driver.

The MO autochanger driver will still be named `autox` on the Series 800. See also “Autochanger Driver and Device Files”.

### Drivers Removed at 10.0

System	Driver	Description
=====		
700	sds	Software Disk Striping.
800	mirror	DataPair.
800	gpio0	CIO GPIO driver.
800	gpio1	NIO GPIO LDM.
800	gpio11	NIO GPIO DM.
800	pdisc	Superseded by cpd (see "New Drivers").
700	s2tape	SCSI-2 tape (functions taken over by scsi_tape driver).

## Autochanger Driver and Device Files

Autochanger driver behavior has changed. As of 10.0, you can mount only as many platters as there are actual autochanger drives, and HP-UX creates only as many device files as are needed to communicate with those drives. In addition, hard-partitioning on autochanger surfaces is no longer supported, and the device-file naming convention has changed, as explained under “Device File Naming Convention” earlier in this section.

9.x autochanger device files will be removed during the upgrade to 10.01.

Because there have been major changes in the 10.0 autochanger driver, HP recommends that you consult the manual *Installing and Administering Optical Jukeboxes on HP 9000 Series 700 and 800 Systems*, HP part number 5960-7624 (Edition 3) to determine what changes, if any, you will need to make to your applications.

**Autochanger Virtual Mounting.** The single largest change in autochanger functionality is that virtual mounting is no longer supported. Specifically, with the 10.0 `ssrfc` driver, you can have only as many surfaces simultaneously mounted (open) as there are drives in the autochanger. Additionally, the “A” and “B” sides of the same cartridge cannot be opened at the same time using the `ssrfc` driver.

The 9.x driver allowed you to virtually mount any or all of the surfaces in the autochanger, and would handle the details of physically mounting surfaces in drives when they were accessed for I/O. With `ssrfc`, when a surface is opened, it is placed in a drive if possible. If no drives are available, the process requesting the surface will block (sleep) unless the request is performed with an `O_NDELAY` (nonblocking) flag, in which case the open request will fail with `EBUSY`.

This change may affect scripts or applications that invoked the 9.x `autoch` driver, and now need to invoke the 10.0 `ssrfc` driver.

## Series 700 SCSI Subsystem and Disk Driver

The HP-UX 10.x Series 700 SCSI Subsystem and Disk Driver have been converged with the Series 800 SCSI Disk Driver in the following ways:

### ■ Default Immediate Reporting behavior

Some new Series 800 systems are capable of using `sdisk`, the SCSI disk driver primarily designed for the Series 700. **Immediate Reporting**, or **WCE** (Write Cache Enable) behavior is now controlled by a new tunable parameter, `default_disk_ir`. This parameter has a profound effect on file system (and raw disk) performance and, conversely, on data integrity if the drive is powered down.

**Immediate Reporting** can be turned either **ON** (set to 1) or **OFF** (set to 0). If `default_disk_ir` is not explicitly set in the `/stand/system` file used to create the kernel, it is assumed to be **OFF** (0).

On new systems, systems on which you “cold install” 10.01, and systems you upgrade from 9.x to 10.01, `default_disk_ir` is set to **ON** (1) on Series 700s and **OFF** on Series 800s.

---

### Note

A disk mounted on a 10.x Series 800 and subsequently moved to a Series 700 system will continue to exhibit **Immediate Reporting OFF** behavior. You can use the `scsictl(1M)` command to turn on **Immediate Reporting**.

This behavior could change in a later release.

---

With **Immediate Reporting** set to **ON**, disk drives which have data caches complete writes when the data is cached, rather than completing once the data is on the disk. This may improve write performance, especially for sequential transfers. But cached data may be lost if a device powerfailure or reset occurs before the device moves the data to the disk. On Series 800 systems, HP recommends that you set this value to **OFF** (0).

`default_disk_ir` also affects delayed-write versus write-through file system behavior.

### ■ Support for LVM High Availability features



The disk driver can return error codes for which the Logical Volume Manager (LVM) has recoverability mechanisms. This behavior is similar to that of the Series 800 disk drivers.

### **Series 700/800 Tape Driver**

The HP-UX 10.x tape driver incorporates into one common set of options the configuration options previously offered by only the Series 700 or 800 driver (with the exceptions listed under “Tape Driver Differences on the Series 700 and 800” later in this section).

Other changes:

- Configuration bits redefined.

The new scheme includes a configuration Property Table containing an array of user-definable configuration choices. You can invoke the Property Table as an option through the minor number.

- Major numbers changed.

- Series 700:

Major numbers 54 and 121 are not supported on 10.x. The new major number 205 provides the configuration options previously available through 54 or 121.

- Series 800:

Major number 212 replaces major number 5.

- Naming convention.

10.x tape device-file names reflect the configuration options now available in the minor number.

For information on configuring tape drives, see the *Configuring HP-UX for Peripherals* manual and the 10.01 `mt(7)`, `scsi_tape(7)`, `mksf(1M)`, `insf(1M)`, `rmsf(1M)`, and `lssf(1M)` manpages.

The new conventions for device-file naming and major and minor numbers are discussed earlier in this section: see “Device File Naming Convention” and “Major and Minor Numbers”.

### **Tape Driver Differences on the Series 700 and 800.**

- Series 700 options not available on Series 800 (except Model K):
  - Partitioning.
  - Fixed Block mode.
- Series 800 modes not available on Series 700:
  - RTE.
  - Suppress Messages.

The driver also differs between Series 700 and 800 systems in its approach to devices it does not recognize.

On the Series 700, the driver will attempt to open any tape device, even if it is a model it does not recognize; whereas on the Series 800, the driver will try to open only the devices that are on its list.

This means that the list of devices that will work with the Series 800 driver is fixed, but the Series 700 list includes devices that “might work” but have not been certified as supported by Hewlett-Packard.

See the *Configuring HP-UX for Peripherals* manual for more information.

### **How This Affects Your System**

**Drivers May Be Removed.** When you upgrade to 10.01, the upgrade program will remove non-HP drivers (and HP drivers in filesets currently not loaded). The 9.x `dfile` or `gen` file that includes the drivers is saved in `/stand/system.old`. The 9.x `/etc/master` file is saved in `/etc/upgrade/save/hostname/etc/master`.

The upgrade program will also remove the device files associated with these drivers. (You may still be able to use in-house and third-party drivers; see “Unmodified Drivers”.)

## Preparing for I/O Convergence

Before upgrading from a 9.x to a 10.x system, do the following:

- Contact application vendors and in-house developers to make sure that any non-HP drivers you depend on will port smoothly to 10.01.
- Use the tools supplied in the Upgrade Preparation Media Tools package to identify references to 9.x device file names (except system default names such as `/dev/lp`, which are still recognized on 10.01 systems), and calls to `mknod` that use major and minor numbers that are invalid on 10.01. Replace these with calls to `mksf` or `insf`.

For details, see the *Upgrading from HP-UX 9.x to 10.x* manual.

- If you are using *DataPair/800*, you must migrate to *MirrorDisk/UX* before you upgrade.

*MirrorDisk/UX* is an optional product that runs under LVM. For instructions on migrating from *DataPair/800*, see Chapter 8, “Managing Logical Volumes,” in the 9.0 Series 800 version of the *HP-UX System Administration Tasks* manual.

---

## Kernel Convergence

HP-UX 10.0 introduced a common set of kernel generation tools and procedures for the Series 700 and Series 800 platforms. These are based on the Series 700 model; the Series 800 model is no longer supported.

### Major Changes for Kernel Convergence

- There is a new utility, `mk_kernel(1M)`, for building the kernel executable file from the source file `/stand/system`.

HP recommends that you use `mk_kernel` (or SAM) to build the kernel on both Series 700 and Series 800 systems.

- The following Series 800 commands are no longer supported:

- `regen`
- `uxgen`

- The Series 800 `genfile` format is no longer supported.

The Series 700 `dfile` format (slightly modified; see the entry for “`config(1M)`” in Chapter 7) is now common to both the Series 700 and 800 platforms, as is `config(1M)` and the new command `mk_kernel(1M)`.

Kernel template files (such as those supplied under `/etc/conf` on a 9.x Series 700 system) are not supplied with 10.x.

- The locations, and in some cases the structuring, of kernel files have changed to conform to the 10.0 file system layout:

- The compiled kernel is now `/stand/vmunix` instead of `/hpux`.
- The kernel source file is now `/stand/system`.
- Kernel libraries and headers have moved from `/etc/conf` to `/usr/conf`.
- The `/etc/master` file and `space.h` files have been replaced by directories, `/usr/conf/master.d` and `/usr/conf/space.h.d` respectively.
- The kernel customize scripts (used for system update) have been broken out from `/system/fileset_name/customize` and moved to `/var/adm/sw/product_name/fileset_name/*`.

For more information, see “The HP-UX 10.0 File System Layout” earlier in this chapter.

## How This Affects Your System

When you upgrade to 10.x, the upgrade process will do the following:

- Extract a new kernel source file `/stand/system` from the running kernel.  
SD will ignore anything in the 9.x `dfile` or `genfile`. The new `/stand/system` file will be in `dfile` format; you cannot generate a 10.x kernel from a file in `genfile` format.
- Save any changes you have made to the 9.x `/etc/master` file in files in the `/usr/conf/master.d` directory.  
But some of your changes may not be reflected in the new kernel. For example, non-HP drivers will not be bound in. See “Drivers”.
- 9.x HP-UX “core” drivers will be replaced in the kernel by new, 10.x-style drivers.
- 9.x kernel libraries will be removed from the system.

## Preparing for Kernel Convergence

Before you begin the upgrade to 10.x, you need to:

- Understand and prepare for the I/O changes that will affect what is included in the 10.x kernel built for you during the upgrade.  
See “Preparing for I/O Convergence” and the corresponding section in the *Upgrading from HP-UX 9.x to 10.x* manual.
- Back up your kernel source file (`/etc/conf/dfile` or `/etc/conf/gen/S800`) and executable (`/hp-ux`).

There are more detailed instructions in the *Upgrading from HP-UX 9.x to 10.x* manual.

Once you are up and running on 10.x, follow the directions in the *HP-UX System Administration Tasks* manual and in the *Configuring HP-UX for Peripherals* manual when you need to do something (such as adding a peripheral or tuning parameters) that involves rebuilding the kernel. Such tasks can usually be done via SAM, the menu-driven system administration tool. SAM is a particularly good option when you first come up on 10.01

and have a lot to get used to in the way of new tools, names, and directory structures.

---

## Software Distributor

The distribution of HP-UX software was completely revised at HP-UX 10.0. HP OpenView Software Distributor 2.0 is used to install, update, remove, and package HP-UX software. This product is referred to as **SD**, or sometimes SD-UX. All HP-UX 10.x software will be shipped on media compatible with SD. SD media cannot be read by installation tools from previous HP-UX releases, nor can you use previous installation tools with HP-UX 10.x software.

### Reasons for the Change

The concept of installing, configuring, removing, and otherwise managing software has not changed. However, the set of commands and some of the capabilities that SD brings to HP-UX are all new. SD provides enhanced functionality, added features, and greater ease of use than that provided by previous installation tools.

### 10.01-Specific Changes to SD

- Upgrade support

The process of updating an HP-UX 9.x system to 10.01 is supported with the use of SD. Some modifications have been made to SD code to detect that it is being used to perform an upgrade.

- NFS Diskless support

Support for installation, configuration, removal, verification, and other software management activities on an NFS diskless cluster is included in 10.01. The new `swcluster` command installs and removes software from diskless clusters. Some modifications to `swinstall` and `swremove` have been made to support this functionality. SAM can launch SD processes to enable software management on an NFS Diskless cluster.

- Protected software installation

Some changes have been made to the information that is required from you during `swinstall` or `swcopy` from “protected” CD-ROM media. These changes are intended to make the installation of such software easier.

- Installation support

- Journaled File system (VxFS) support—non-root/boot only
- Full support for initial system installation onto legacy systems
- 9-track media verification

### **Overview of SD Commands**

Table 4-2 provides a brief overview of basic SD commands and how they map to corresponding 9.x commands:



**Table 4-2. 9.x Commands versus SD Commands**

Purpose	9.x Commands	SD Commands
Initial software installation	<code>install</code>	<code>swinstall(1M)</code>
Update existing software	<code>update</code>	<code>swinstall(1M)</code>
Remove software	<code>rmfn</code>	<code>swremove(1M)</code>
Perform auxiliary installation or removal actions	customize scripts	control scripts
Create media for use with <code>swinstall</code>	<code>fpkg</code>	<code>swpackage(1M)</code>
Distribution daemon	<code>netdistd</code>	<code>swagentd(1M)</code>
Distribution structure	netdist server area	depot
Create or update a software distribution server structure	<code>updist</code>	<code>swcopy</code>
Process that performs SD command's tasks	no direct equivalent	<code>swagent(1M)</code>
Register an SD depot	no direct equivalent	<code>swreg</code>
Run selected product configure scripts	no direct equivalent	<code>swconfig(1M)</code>
List features, status, and attributes of software entities	no direct equivalent	<code>swlist(1M)</code>
View or modify Access Control Lists	no direct equivalent	<code>swacl(1M)</code>
Modify the Installed Product Database	no direct equivalent	<code>swmodify(1M)</code>
Verify the correctness of an installed product	no direct equivalent	<code>swverify(1M)</code>

---

**Note** Hewlett-Packard does not support the creation of bundles at this time.

---

In addition, you can use the `fpkg2swpkg` utility to help you convert `fpkg` Product Specification Files (PSFs) to `swpackage` format. You will still need to make manual changes in all but the most simple cases, but `fpkg2swpkg`

eliminates much of the clerical work involved in these conversions. For more information, refer to the `fpkg2swpkg(1M)` manpage.

## SD Interfaces

SD provides the following three interfaces:

- An interactive graphical user interface available for the `swinstall`, `swcopy`, and `swremove` commands.
- An interactive user interface suitable for ASCII terminals (for Series 800 computers) for the `swinstall`, `swcopy`, and `swremove` commands.
- A non-interactive command-line user interface for all commands.

## Where to Find Additional Information

For additional information on the SD commands and features, refer to the SD commands manpages, the SAM online help, the *Installing HP-UX 10.01 and Updating HP-UX 10.0 to 10.01* manual (B2355-90078), the *Upgrading from HP-UX 9.x to 10.x* manual (B2355-90083), and the *Managing HP-UX Software with SD-UX* manual (B2355-90080).

---

## Bundles

There are three types of bundles-related changes for the 10.01 release:

- New Bundle Changes, Tags, and Definitions for 10.01
- 10.0 Core Bundles Moving to 10.01 Applications Media
- ASE (Asian System Environment) Delivery Re-structuring

## New Bundle Changes, Tags, and Definitions for 10.01

Series 700:

HP0nLineJFS	Online Journalled	(Separately sold product)
	File System	

2UserDegrade HP-UX 2-User License (For degrading the user  
license level)

The only purpose of the 2UserDegrade bundle/fileset is to degrade the user license of a system (for example, when transferring a higher level license to a different system). This bundle contains the UserLicense.02-USER fileset. The 02-USER fileset was removed from the operating environment bundles at 10.01.

Series 800:

The Series 800 Runtime bundles listed below have changed their content and descriptions. These bundles will have the X11, Imaging, ScreenCapture, Audio, and SharedPrint filesets removed to reduce the size of the operating system. Asian versions of these bundles include the X11.MKFONDIR fileset to enable native language printing via NL00.

Tag	Title
---	-----
HPUXDanRT800	Danish non-Graphics Runtime HP-UX Environment
HPUXDutRT800	Dutch non-Graphics Runtime HP-UX Environment
HPUXEngRT800	English non-Graphics Runtime HP-UX Environment
HPUXFinRT800	Finnish non-Graphics Runtime HP-UX Environment
HPUXFreRT800	French non-Graphics Runtime HP-UX Environment
HPUXGerRT800	German non-Graphics Runtime HP-UX Environment
HPUXItaRT800	Italian non-Graphics Runtime HP-UX Environment
HPUXJpnRT800	Japanese non-Graphics Runtime HP-UX Environment with JSE
HPUXKorRT800	Korean non-Graphics Runtime HP-UX Environment with KSE
HPUXNorRT800	Norwegian non-Graphics Runtime HP-UX Environment
HPUXSchRT800	Simplified Chinese non-Graphics Runtime Environment with SSE
HPUXSpaRT800	Spanish non-Graphics Runtime HP-UX Environment
HPUXSweRT800	Swedish non-Graphics Runtime HP-UX Environment with SSE
HPUXTChRT800	Traditional Chinese non-Graphics Runtime HP-UX Environment with TSE

The Series 800 10.0 HPUX\*\*\*RT800 bundle tags are being changed to HPUX\*\*\*GS800, and the titles are being changed from “Runtime” to “Runtime with Graphics.”

Tag	Title
---	-----
HPUXDanGS800	Danish HP-UX Runtime with Graphics
HPUXDutGS800	Dutch HP-UX Runtime with Graphics
HPUXEngGS800	English HP-UX Runtime with Graphics
HPUXFinGS800	Finnish HP-UX Runtime with Graphics
HPUXFreGS800	French HP-UX Runtime with Graphics
HPUXGerGS800	German HP-UX Runtime with Graphics
HPUXItaGS800	Italian HP-UX Runtime Environment
HPUXJpnGS800	Japanese HP-UX Runtime with Graphics
HPUXKorGS800	Korean HP-UX Runtime with Graphics
HPUXNorGS800	Norwegian HP-UX Runtime with Graphics
HPUXSchGS800	Simplified Chinese HP-UX Runtime with Graphics
HPUXSpaGS800	Spanish HP-UX Runtime with Graphics
HPUXSweGS800	Swedish HP-UX Runtime with Graphics
HPUXTChGS800	Traditional Chinese HP-UX Runtime with Graphics

The Integrated Logon fileset is being removed from Operating Environment bundles to reduce the size of the operating system. However, it will still be available on the core media as a user-selectable (HPUXAdditions) bundle:

Tag	Title
---	-----
Integ-Logon	Integrated Logon Bundle

The Online Journalized Filesystem will be available as a bundle (purchasable product) on the Application Media:

Tag	Title
---	-----
B3928AA	HP OnLineJFS
2User Degrade	HP-UX 2-User License (For degrading the user license level)

## 10.0 Core Bundles Moving to 10.01 Applications Media

The following bundles will not be changing except that at 10.01 they will be delivered on the Applications media rather than the Core media as they were at 10.0.

Tag	Title
---	-----
(S700)	
B3884CA_AGL	HP-UX 8-User License
B3884CA_AGN	HP-UX 32-User License
(S800)	
B3897A_AGL	HP-UX 8-User License
B3897A_AGM	HP-UX 16-User License
B3897A_AGN	HP-UX 32-User License
B3897A_AGP	HP-UX 64-User License
B3897A_AGS	HP-UX Unlimited-User License
B3919AA_AGL	HP-UX 8-User License
B3919AA_AGM	HP-UX 16-User License
B3919AA_AGN	HP-UX 32-User License
B3919AA_AGP	HP-UX 64-User License
B3919AA_AGS	HP-UX Unlimited-User License

## ASE (Asian System Environment) Delivery Re-structuring

For 10.0, the Asian Operating Environments consisted of the combination of a localized bundle from the core media and an ASE bundle from separate pieces of media. For 10.01, new bundles have been developed that integrate the two previous bundles into a single bundle that allows you to install all necessary components via a single bundle. This allows Asian versions of HP-UX to have the same level of operating system integration as European versions had at 10.0. These new structures apply to both the S700 and S800 platforms. The new bundles are:

Series 700:

HPUXJpnDT700	Japanese HP-UX Desktop Environment	(with JSE)
HPUXJpnRT700	Japanese HP-UX Runtime Environment	(with JSE)

HPUXKorDT700	Korean HP-UX Desktop Environment	(with KSE)
HPUXKorRT700	Korean HP-UX Runtime Environment	(with KSE)
HPUXSChDT700	Simplified Chinese HP-UX Desktop Environment	(with SSE)
HPUXSChRT700	Simplified Chinese HP-UX Runtime Environment	(with SSE)
HPUXTChDT700	Traditional Chinese HP-UX Desktop Environment	(with TSE)
HPUXTChRT700	Traditional Chinese HP-UX Runtime Environment	(with TSE)

Series 800:

HPUXJpnRT800	Japanese HP-UX Runtime without Graphics	(with JSE)
HPUXJpnGS800	Japanese HP-UX Runtime with Graphics	(with JSE)
HPUXKorRT800	Korean HP-UX Runtime without Graphics	(with KSE)
HPUXKorGS800	Korean HP-UX Runtime with Graphics	(with KSE)
HPUXSChRT800	Simplified Chinese HP-UX without Graphics	(with SSE)
HPUXSChGS800	Simplified Chinese HP-UX with Graphics	(with SSE)
HPUXTChRT800	Traditional Chinese HP-UX without Graphics	(with TSE)
HPUXTChGS800	Traditional Chinese HP-UX with Graphics	(with TSE)

The structure of bundles on media will also change from 10.0. At 10.0, the media was structured as follows:

- S700 CORE MEDIA—All European language Operating Environments and “Asian Operating Environment Bases” for S700 (Included with ABA, ABJ, AB0, AB1, and AB2 media options)
- S800 CORE MEDIA—All European language Operating Environments and “Asian Operating Environment Bases” for S800 (Included with ABA, ABJ, AB0, AB1, and AB2 media options)
- JSE MEDIA—Japanese System Environment (JSE) bundles for both S700 and S800 platforms (Included with ABJ media option only)
- KSE MEDIA—Korean System Environment bundles (KSE) for both S700 and S800 platforms (Included with AB1 media option only)
- SSE MEDIA—Both Simplified and Traditional Chinese System Environment bundles (SSE/TSE) for S700 and S800 platforms (Included with AB0 and AB2 media options only)

At 10.01 the Asian bundles are delivered as follows:

#### **4-66 Major Changes for HP-UX 10.0 and 10.01**

- S700 CORE MEDIA—All European language Operating Environments for S700 and all core Asian localized filesets, but no Asian Operating Environment bundles (Included with ABA media option only)
- S700 JSE MEDIA—All European language Operating Environments for S700, all core Asian localized filesets, and Japanese Operating Environment bundles complete with JSE (Included with ABJ media option only)
- S700 KSE MEDIA—All European language Operating Environments for S700, all core Asian localized filesets, and Korean Operating Environment bundles complete with KSE (Included with AB1 media option only)
- S700 SSE MEDIA—All European language Operating Environments for S700, all core Asian localized filesets, and Simplified and Traditional Operating Environment bundles complete with SSE/TSE (Included with AB0 and AB2 media options only)
- S800 CORE MEDIA—All European language Operating Environments for S800 and all core Asian localized filesets, but no Asian Operating Environment bundles (Included with ABA media option only)
- S800 JSE MEDIA—All European language Operating Environments for S800, all core Asian localized filesets, and Japanese Operating Environment bundles complete with JSE (Included with ABJ media option only)
- S800 KSE MEDIA—All European language Operating Environments for S800, all core Asian localized filesets, and Korean Operating Environment bundles complete with KSE (Included with ABK media option only)
- S800 SSE MEDIA—All European language Operating Environments for S800, all core Asian localized filesets, and Simplified and Traditional Operating Environment bundles complete with SSE/TSE (Included with AB0 and AB2 media options only)

---

## Known Problems and Workarounds (for 10.01)

This section contains known problems that *had not* been fixed for 10.01 and takes the place of the Software Status Bulletin (SSB) that has been provided with previous releases. For information on problems that *have* been fixed, refer to the Software Release Bulletin (SRB) located on the 9.x-based HP-UX Upgrade Tools for 9.x to 10.x package in `/etc/newconfig/10RelNotes/10.01SRB`. For 10.10 known problems and workarounds, see the section that documents the topic.

### **hpterm**

Problem:

The version of **hpterm** included with this release contains a bug that prevents it from exporting data of type “string” when responding to a Paste operation. It exports type “compound text” instead.

Workaround:

Since most clients handle “compound text” type, this is not usually a problem. However, FrameMaker does not support the “compound text” type. Therefore, do not paste from **hpterm** to FrameMaker until this bug is fixed.

### **fbackup**

Problem:

**fbackup** fails to write to an HPIB DDS device on 9.0 on an 832 machine when a high density device file is used. However, no error occurs with a medium density or high density device file when **tar** is used. If an 835 (CIO) machine is used, no error occurs.

Workaround:

Reel-to-reel tapes devices use low, medium, and high density device files. As of 10.0, these terms have been replaced with a new naming convention. You can no longer specify a “high” density for a DDS device. Thus, it is quite likely that this problem no longer exists in 10.x.



## **EISA PSI Card ID**

Problem:

During the first boot-up session after you install and automatically configure an EISA card, `ioscan` does not display the EISA ID associated with the newly installed card. (This functionality was new for 10.0; the 9.x version of `ioscan` did not print out the EISA card ID).

Workaround:

To display the EISA ID, reboot your system a second time after `eisa_config` automatically configures the PSI card after installation.

## **swapon**

Problem:

The `swapon(1M)` command fails due to having no extra `maxswapchunks` available for additional swap. Then the `lv` is left open, and it is not possible to use `lvremove(1M)` in order to delete the `lv`.

Workaround:

The system must be reconfigured with a larger value for the tunable `maxswapchunks` and rebooted in order to enable all of the additional swap requested.

## **Unresponsive Small-Memory System**

Problem:

A set of processes can hang when running disk-intensive loads on a small-memory system. The system as a whole may become unresponsive.

Workaround:

The following factors are relevant:

- File systems with a large (64 KB) block size. To check this, as `root` enter:

```
# dumpfs <special> | grep ^bsize
```

- A small lower limit (5 percent or less) for the tunable parameter `dbc_min_pct` in the system file (`/stand/system`). The “small system” configuration sets this to 2 percent.

This is really a symptom of a mis-configured system. The workaround is to `regen` the kernel using a larger lower bound for the buffer cache. A conservative formula for the number of pages is:

```
number-simultaneous-processes
  times
  (largest-filesystem-block-size / 4096)
  times
  4
```

For example, with 25 simultaneous active processes and a 64 KB file system:

```
25 * 16 * 4 ==> 1600 pages ==> 6 Mbytes
```

Express this as a percentage of physical memory for `dbc_min_pct`.

## Watchdog Timer

Problem:

When booting, if a device does not respond or manages to hang the DAM, then a watchdog timer pops and prints the following:

```
"I/O hardware probe timed out at path X.Y.Z"
```

The problem is that the path points to the last successfully probed device and not the currently failing device.

Workaround:

There is no workaround at this time.

## swremove

Problem:

Local language files are not removed by the `swremove` program.

Workaround:

## 4-70 Major Changes for HP-UX 10.0 and 10.01

You can delete these files yourself if you do not want them. They are in the `/usr/lib/nls` directory.

## **Kermit**

Problem:

Kermit will leave RTS (HW Flow Control) turned on after exiting. This may negatively impact other uses of the serial port.

Workaround:

Avoid using the hardware flow control device while using Kermit.

## **uucp**

Problem:

`uucp` will not copy the file to a final destination when `system_name!system_name! ... !system_name!path_name` is used as the destination.

Workaround:

The copy will work if you do it point-to-point.

## **uucico**

Problem:

`uucico` will not work correctly if you use a baud rate higher than 19200.

Workaround:

Use a lower baud rate.

## **uugetty**

Problem:

Any application that uses `uugetty` will not function properly if a baud rate higher than 19200 is used.

Workaround:

1. Use a lower baud rate.
2. Use `getty` instead of `ugetty` by making the following change:  
In `/etc/inittab`, change all references to `/usr/lbin/ugetty` to `/usr/sbin/getty` for ports running at speeds of 38,400 baud or higher.

### **hpux(1M)**

Problem:

The `hpux(1M)` restore utility fails if the tape and disk are not on the same bus.

Workaround:

None. This problem existed in 9.x.

### **ftp(1)/ftpd(1M)**

Problem:

If you log into a 10.0 or later system using `ftp` in conjunction with NIS served by a 9.x system, the login will fail without additional configuration.

NIS gets a `passwd` entry from its server system, which includes a login shell. When the NIS server is a 9.x system, the shell returned will have a 9.x path (`/bin/ksh`). `ftpd` requires that the shell for a given user be in the list returned by `getusershell(3C)`. This list is the contents of `/etc/shells`, or, if that file does not exist, the following default list:

```
/sbin/sh
/usr/bin/sh
/usr/bin/rsh
/usr/bin/ksh
/usr/bin/rksh
/usr/bin/csh
/usr/bin/keysh
```

This is a list of the shells' 10.x paths, not 9.x paths.

Workaround:

Create an `/etc/shells` file, adding the above list and the 9.x path of any shells that you want to be allowed login access.

## **4-72 Major Changes for HP-UX 10.0 and 10.01**

## **MPower**

Problem:

HP MPower does not run on HP-UX 10.x. If MPower is detected on a system that is being upgraded from HP-UX 9.x to HP-UX 10.x, you will see a warning message that MPower will be disabled if the upgrade continues.

Workaround:

You can chose whether to continue the upgrade or not. Some files may be left in their old locations, but the results are unpredictable if you try to use them on an HP-UX 10.x system.

A separate product providing complete MPower 2.0 functionality for HP-UX 10.x systems will be available later.

## **SLIP**

Problem:

In a dial-in with login environment, the command line `"" /usr/bin/ppl\s -i\host running` (as configured in the `ppl.remotes` file) may fail to be executed by the dial-in host, if the login shell is `sh` or `ksh`.

Workaround:

Refer to the “SLIP” section in Chapter 5 for workarounds.

## **extendfs(1M)**

Problem:

If you try to extend the Journaled File System (VxFS) using the `extendfs` command and you do *not* have HP OnlineJFS, you may not be successful.

Workaround:

Contact your Support Representative or HP SupportLine for a possible workaround.



## Other Operating System and Subsystem Changes

---

This chapter covers the following topics:

- Address Space Layout
- Commercial Security
- Curses
- **DCE (Distributed Computing Environment)**—*changed at 10.20 and 10.10*
- Diagnostics—*changed at 10.10*
- Dynamic Buffer Cache
- Fast Symlinks
- File System Changes—*changed at 10.10*
- Graphics—*changed at 10.20 and 10.10*
- HP Process Resource Manager (HP PRM)
- HP Support Watch and HP Remote Watch
- Instant Ignition
- Kermit
- Kernel Configuration—*changed at 10.10*
- Kernel Tunables—*changed at 10.10*
- MC/ServiceGuard
- Measurement Systems (KI/MS)
- Memory
- **Multimedia for 10.20**—*changed at 10.20*
- Multimedia for 10.10—*changed at 10.10*
- Peripheral Devices—*changed at 10.10*
- POSIX Realtime Extensions
- **Programming Languages**—*changed at 10.20 and 10.10*
- **SAM (System Administration Manager) for 10.20**—*changed at 10.20*
- SAM (System Administration Manager) for 10.10—*changed at 10.10*
- SAM (System Administration Manager) for 10.0—*changed at 10.0*
- SCSI Pass-Through for Series 800 NIO Bus
- **SharedPrint/UX**—*changed at 10.20 and 10.10*

- **SharedX**—*changed at 10.20*
- Shells
- Small Disks and Memory
- System Calls
- **typedef changes**—*changed at 10.20*
- UUCP—*changed at 10.10*
- **VUE (HP VUE 3.0)**—*changed at 10.20 and 10.10*
- **X Windows**—*changed at 10.20 and 10.10*

---

## Address Space Layout

As of HP-UX 10.0, `ld(1)` can generate Series-700-style `EXEC_MAGIC` executables, which allow greater private address space, on Series 800 computers as well as Series 700. See the `ld(1)` manpage under `-N` for more information on `EXEC-MAGIC`.

---

## Commercial Security

10.x commercial security replaces the shadow `passwd` feature with new, more stringent security features.

### New Features

- Password management

You can set password length (up to 40 characters), delay between logins, maximum number of unsuccessful tries, lock account, etc.

- Time-based access control.

You can restrict a user's login to certain times of the day or days of the week.

- Location-based access control.

A user's login can be restricted to a particular terminal or host.

## 5-2 Other Operating System and Subsystem Changes



## SVR4: getspent(3C)

`getspent(3C)` and the related calls `getspnam(3C)`, `endspent(3C)`, and `setspent(3C)` provide SVR4-conformant programmatic read access to shadow password-like entries for each user. They can be invoked only by a process that has superuser privileges.

`setspent()` and `endspent()` reset and end access to entries. `getspent()` returns the next entry, and `getspnam()` returns a particular entry based on login name.

These calls allow you to port code that uses SVR4-style shadow password files for reads. There are no equivalent routines for writes.

If you do not need SVR4 conformance, you can use `getspwent` and related routines instead, but you will need to link with `libsec`; see “Shadow Password Routines” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

## New Commands, Calls and Files

Commercial Security has introduced a number of changes. Once you have 10.x on your system, see the manpages for details.

```
bigcrypt(1)
getprdfent(3)
getprpwent(3)
getprdvagent(3)
getprtcent(3)
iscomsec(3)
default(4)
prpwd(4)
devassign(4)
ttys(4)
authcap(4)
```

(See also `lckpwwdf(3C)`, `ulckpwwdf(3C)` in “New libc Routines at 10.0” in Chapter 7.)

## Old Features No Longer Supported

- `/.secure/etc/passwd` (shadow password)

Applications accessing the shadow `passwd` file *directly* no longer worked as of 10.0.

`getspwent(3)`, which relies on the shadow password file, has been rewritten to get its information from the new 10.x protected password database, and its external interface has not changed. Applications that accessed the shadow password file via this library call will still work, but they will need to be relinked on 10.x using `libsec`.

---

### Note

The shadow password routines that are in `libc` on 10.x no longer provide any functionality. They are stubs that allow applications that were linked with 9.x shared libraries to continue to work on 10.x non-trusted systems. 9.x executables that call these routines *will not* work on 10.x trusted systems, or on any 10.x system, trusted or non-trusted, if they were linked archived.

See “Shadow Password Routines” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual for details.

---

- The `fggetspwent(3)` and `putspwent(3)` calls.

These routines still exist in 10.x; they have been moved to `libsec`. But applications using them to read or update `/.secure/etc/passwd` will fail because the file no longer exists. See the 10.x `getprpnam(3)` manpage for replacement routines.

## Commands, Calls, and Files Changed

Commercial Security has been modified. Once you have 10.x on your system, check the manpages for details.

<code>login(1)</code>	- added security behavior
<code>passwd(1)</code>	- added security behavior
<code>su(1)</code>	- added security behavior
<code>passwd(4)</code>	

### 5-4 Other Operating System and Subsystem Changes

```
init(1M)          - added security behavior
getspwent(3)     - removed /.secure/etc/passwd processing
getpwent(3)      - see below
```

`getpwent(3)` routines will no longer return the password, audit ID, and audit flag in the password structure.

The interface remains the same, the returned structure remains the same, but the information is not returned. To obtain password and audit information, use `getprpwent(3)`

## Password and Password Aging

If your 9.x trusted system has password aging, you should turn it off before upgrading to 10.01.

HP-UX 10.x uses a new password aging scheme for trusted systems. If a user account has no password aging, it will automatically use the system default, which is stored in the `/tcb/files/auth/system/default` file. The default values are a password lifetime of 28 weeks, and an expiration time of 26 weeks. You can change these values via SAM.

Users can also use SAM to set their own password aging. Individual password aging values are stored in the new 10.x protected password database, which contains a password file for each user.

The aging cycle for all passwords will be restarted when you upgrade a trusted system to 10.01.

The system default file does not allow null passwords. If any account has a null password, you must use SAM to provide one after you upgrade to 10.01.

Use `pwck -s` before upgrading to 10.01 to identify potential problems in `/.secure/etc/passwd`.

These changes affect only trusted systems; on non-trusted systems, any password aging established in `/etc/passwd` will remain in effect after you upgrade to 10.01.

## Differences Between 9.x and 10.01

Your system will upgrade to 10.01 with security automatically turned on if the 9.x system is already secured.

You will notice the following changes on 10.x (see the previous topics for more details):

- `/.secure/etc/passwd` is no longer supported.
- Shadow password routines have been moved to `libsec`; the routines in `libc` are stubs.
- `getpwent(3)` will no longer return the password, audit ID, and audit flag in the password structure.

For information on how these changes affect 9.x executables, see “Shadow Password Routines” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

When you secure a 10.x system (by choosing the security option in SAM), the following will happen:

- The passwords will be removed from `/etc/passwd` and replaced by `*`, as in 9.x.
- A set of default templates (or databases) will be put in place.

`/tcb/files/auth/system/default`

This is the system default file for each user and terminal.

`/tcb/files/ttys`

This is the terminal information file.

`/tcb/files/devassign`

This is the device information file. The only device supported in 10.x is the terminal. The only field is the list of users allowed on a given terminal.

- A file will be created for every user in `/etc/passwd`.

This file contains the user’s encrypted password, along with other information such as password aging, number of unsuccessful tries allowed, maximum password length allowed, times of day the user can log in, and so on.

## 5-6 Other Operating System and Subsystem Changes

The files are stored as:

```
/tcb/files/auth/[a-z,A-Z]/*
```

For example, information for user `jane` would be stored in `/tcb/files/auth/j/jane`

This file is also known as the protected password database.

### Limitations

- A 10.x secure system will not work in an NIS domain.

If placed in an NIS environment, a secure system could prevent everyone from accessing the secure system.

- A 10.x secure system can coexist with the Distributed Computing Environment (DCE).

After logging in to the secured system a user can perform another `dce_login` and have access to DCE applications. But if DCE is explicitly installed on a secured system, then the login and password commands will be replaced by DCE versions that do not recognize the security attributes. In this case, system errors could occur.

---

## Curses

*For 10.0:*

- Curses-Color product.

There is a new product called Curses-Color. Refer to “Curses-Color” for details on the functionality of, and porting to, the color curses library.

- Curses library.

The HP-proprietary curses library (`-lcurses`, `-ltermcap`, and `-ltermlib`), now called the V.4 Curses Library, will not be supported in future HP-UX releases. This library will be replaced by a COSE-compliant library to support maximum portability of applications. To prepare for this change, you should develop all new curses applications with the currently available

color curses library `-lcur_colr`, and port all existing curses applications to the new color curses library.

If you absolutely cannot move forward to the color curses library, and need to be able to relink your application, use the `-lHcurses` library or use the `-L/usr/old/usr/lib` option. Linking against this library or directory ensures binding with the HP-proprietary library. Additionally, if you need to recompile, then use `-I/usr/old/usr/include` (the curses header files under `/usr/include` will be replaced with the COSE-compliant header files).

- Files moved to conform to the 10.0 file system layout.

Description	Old Location	New Location
Library	<code>/lib/libcurses.sl</code>	<code>/usr/lib/libcurses.sl</code>
File	<code>/etc/termcap</code>	<code>/usr/share/lib/termcap</code>
Database	<code>/usr/lib/terminfo</code>	<code>/usr/share/lib/terminfo</code>

(The `termcap` file is delivered in `/usr/newconfig/usr/share/lib/termcap`; the curses installation script moves it to `/usr/share/lib/termcap`.)

- Four routines added to conform to XPG3.

```
def_shell_mode
def_prog_mode
reset_shell_mode
reset_prog_mode
```

- New fileset for `terminfo` files.

Terminal Type	Old Fileset	New Fileset
VT 100	NONHPTERM	UX-CORE
Wyse 60	NONHPTERM	UX-CORE
Xterm	NONHPTERM	UX-CORE

- Minicurses is no longer supported.
- HP-15 and EUC multibyte are not supported.
- In previous releases of HP-UX, when `nonl()` is called within an application, newline is translated into a return and line feed on output. With 10.01, `nonl()` works correctly, and newline is translated into a line feed on output. The fix has been made to the HP V.4 Curses Library, but the bug exists in

## 5-8 Other Operating System and Subsystem Changes

10.0 and in USL curses. Therefore, having the fix will break all applications that depend on `nonl()` not working. Applications that depended on the earlier `nonl()` functionality should no longer reference it.

## **Curses-Color**

Curses-Color is a COSE-compliant color-management package. A later release will provide the next step in COSE compliance, support for multibyte Application Programmer Interfaces (APIs).

In addition to color, Curses-Color provides some 260 APIs, rather than the 135 offered by HP-UX Curses.

Curses-Color affects `curses` routines that:

- Add character strings to a `curses` window.
- Manipulate the `curses` window background.
- Create `curses` borders.
- Provide access to various low-level `curses` functions.
- Provide interfaces to the `terminfo` database.
- Create `curses` windows.
- Control refresh.
- Read or write a `curses` screen from or to a file.
- Handle soft labels.

The product consists of two filesets, `CURSES-MIN` and `CURSES-PRG`.

### ■ `CURSES-MIN`:

Contains the shared library and related commands and a minimal set of `terminfo` files.

You must load `CURSES-MIN` in order to use Curses-Color.

### ■ `CURSES-PRG`:

Contains the archived library and the header files `curses.h` and `term.h`.

You need to load `CURSES-PRG` only if you are going to compile and link programs using Curses-Color.

### ■ Compile instructions:

Your `makefile` must supply the include path and library path as follows:

```
-I /usr/include/curses_colr
-l cur_colr
```

- **Commands:**

The Curses-Color commands are: `tic_colr`, `infocmp_colr`, `captoinfo_colr`, and `tput_colr`.

- **terminfo path:**

The `terminfo` database is `/usr/share/lib/terminfo_colr`.

If Curses-Color does not find `terminfo` file there, it looks in `/usr/share/lib/terminfo`.

### **Effect on Applications**

- Existing applications are not affected.

They will continue to run as they did until you decide to incorporate Curses-Color. Do not move or remove the HP-UX `curses` library, `/usr/lib/libcurses.sl`.

- If you modify an application to use the Curses-Color library, you may need to change some calls to `curses` APIs.

About 2% of the current HP-UX APIs have a different interface from the corresponding Curses-Color API.

### **Curses Documentation**

In addition to the manpages for the HP V.4 Curses Library, the following documentation is available. Prentice Hall Publisher's toll-free phone number is 1-800-947-7700.

- *UNIX Curses Explained* (\$44.00)

Author: Berny Goodheart

Publisher: Prentice Hall

ISBN# 0-13-931957-3

This manual is highly recommended for application developers who are new to Curses programming.

## **5-10 Other Operating System and Subsystem Changes**



- *UNIX System V Release 4 Programmer's Reference Manual* (\$45.00)  
Author: AT&T UNIX System Laboratories  
Publisher: Prentice Hall  
ISBN# 0-13-947029-8
- *UNIX System V Release 4 System Administrator's Reference Manual* (\$49.00)  
Author: AT&T UNIX System Laboratories  
Publisher: Prentice Hall  
ISBN# 0-13-947011-5
- *UNIX System V Release 4 System Files and Devices Reference Manual*  
(\$41.00)  
Author: AT&T UNIX System Laboratories  
Publisher: Prentice Hall  
ISBN# 0-13-951302-7
- *Terminal Control: User's Guide*  
HP 9000 Computers  
HP Part No. B1862-90013

This is the HP Curses 9.0 Manual which is provided with the 9.0 documentation set for basic information on Curses.

---

## **DCE (Distributed Computing Environment)**

*For 10.20 and 10.10:*

For 10.20, HP DCE/9000 Version 1.5 client software is included in default HP-UX system software bundles. For 10.10, HP DCE/9000 Version 1.4.1 client software is included in default HP-UX system software bundles.

The functionality ensures that any HP-UX 10.20 and 10.10 system that has installed the default system software bundles can be configured as a DCE client, and as such, will operate with HP/DCE 9000 servers or with DCE servers from another vendor and can run any DCE RPC application.

Also bundled with HP-UX 10.20 and 10.10 is a standalone POSIX Threads library (`libcma.sl`) that enables developers to use the pthreads API (Application Programmer Interface) without linking against the entire DCE library.

---

**Note**

- HP DCE/9000 runs only on a long-name file system. If you have a short-name file system, you must first run `convertfs(1m)` to convert your file system to long names.
  - Hewlett-Packard requires use of the ANSI C compiler when building any threaded application (this includes ALL HP DCE applications.) Hewlett-Packard does not provide support for threaded applications that were not compiled using ANSI C.
- 

DCE software included in HP-UX 10.20 and 10.10 is either “default-bundled” (included in one of the standard Series 700 or Series 800 system software bundles) or “optional-bundled” (available on the media and optionally installable). See “Functionality”, “Default-Bundled Software”, and “Optional-Bundled Software” for more information.

**Functionality**

*For 10.20:*

- HP DCE Measurement Service (DMS) to monitor resource utilization and performance of HP DCE 1.5 servers.
- Support for large UUIDs.
- Support for large files APIs; however, there is no support for creating or operating on large files within DFS.
- Integrated login support for CDE/PAM.
- Support for context-switching 64-bit machine registers in DCE threads (`libcma` and `libdce`).
- Thread-safe wrapper for the new POSIX 1003.1b calls (`waitid`, `nanosleep`, and `setrlimit64`).

In addition, HP DCE Version 1.5 contains numerous bug fixes.

**Impact**

*For 10.20:*

There will be no impact to the general (non-DCE) HP-UX user. Both shared and archived threaded applications will run on both PA1.1 and PA2.0 machines at 10.20 without recompiling.

**5-12 Other Operating System and Subsystem Changes**

## Compatibility

*For 10.20:*

HP DCE 1.5 supports binary compatibility with HP DCE 1.2/1.2.1, HP DCE1.3/1.3.1, and HP DCE 1.4/1.4.1/1.4.2. Applications linked with the archived HP DCE 1.2/1.2.1 or 1.3/1.3.1 `libdce` are fully compatible with applications built with HP DCE 1.5 libraries. These applications can share login contexts and credentials without loss of data.

Binary compatibility for statically-linked HP DCE 1.2/1.2.1/1.3/1.3.1 applications can be disabled, resulting in minor performance gains and slightly smaller credential files. By default, binary compatibility is enabled when HP DCE 1.5 is installed and configured. You may disable binary compatibility on a per-host basis with the following command:

```
dcecp> hostvar set -secbinarycompat off
```

To enable binary compatibility after it has been disabled, do the following:

1. Issue the command

```
dcecp> hostvar set -secbinarycompat on
```

2. Stop and restart DCE daemons.
3. If using Integrated Login, log out, and log in.

If statically-linked HP DCE 1.2/1.2.1/1.3/1.3.1 application purges a login context (via `sec_login_purge_context`) that an HP DCE 1.5 application had created or refreshed, one of the credential files will not be deleted from the disk. This file is located in `/var/opt/dce/security/creds`. The filename will consist of the unique credential cache id associated with the login context with a `data.db` suffix. Administrators can manually remove this file.

For information about the U.S./Canada version of HP DCE, see the *HP DCE/9000 Version 1.5 U.S./Canada Version Release Note*.

Due to the following changes at OSF DCE 1.1, there may be source code incompatibility with previous HP DCE releases. The following changes will impact user programs and scripts:

- Several control programs were replaced with dcecp and several corresponding daemons no longer exist.
- ACL manager facilities were provided withing libdce; it is now necessary to transition user-written ACL managers to use the new facilities.

For detailed information about these items, see the release notes in:

`/opt/dce/newconfig/RelNotes/HPDCE1.5RelNotes.ps`

`/opt/dce/newconfig/RelNotes/HPDCE1.5RelNotes.txt`

*For 10.10:*

- Parity with OSF/DCE Version 1.1
- Single Administrative DCE Control Program - dcecp
- DCE Daemon (combines rpcd and sec\_clientd) - dced
- Cell Aliasing
- Hierarchical Cells (limited)
- Serviceability improvements
- Security Delegation - intermediary servers can operate on behalf of the initiating client while preserving identities and ACLs
- Auditing - tracking of security-related events
- Extended Generic Security Service Application (GSSAPI) - message passing applications can use DCE security (includes GDS)
- Extended Registry Attributes (ERA) - provides single sign-on across non-Unix platforms and legacy systems by enabling a secure way of associating additional security information with users and groups
- Extended Login Capabilities - pre-authentication and password management allows applications to require access only from trusted machines
- Group Override - customizes group name mapping from host to host to allow DCE to adopt to various operating system conventions
- Internationalization Interfaces - message catalogs for all user-visible messages
- Character Code Set Interoperability - allows development of RPC applications that automatically convert character data from one code set to another
- IDL Compiler performance enhancements - smaller stub size and support for a number of new IDL constructs
- RPC Performance Enhancements - allows additional client sockets during peak usage and optimizes RPC runtime packets
- GDS Enhancements - ease of programming and administration

## **5-14 Other Operating System and Subsystem Changes**

- DFS/NFS Gateway (was available in HP DCE 1.2)
- DFS Delegation - files can be passed with initiator's privileges intact
- Subtree Operations - allows large-scale administrative name changes within cells
- Distributed Time Server (DTS) Enhancements - remote administration
- Enable single threaded Datagram protocol clients
- Single cds client process per machine
- Support for Kerberos V5 clients
- dced/dcecp time synchronization
- Increase performance of local RPC
- IDL++ user defined exceptions
- Compatibility with ServiceGuard
- Reduced disk and memory consumption for client host machines

### **Default-Bundled Software**

DCE default-bundled software is included in one or more of the Series 700 Runtime, Series 700 Desktop, Series 800 Server, and Series 800 Minimum Functionality bundles.

DCE default-bundled software includes:

- DCE Shared Libraries:
  - pthreads shared library (`libcma.sl`)
  - International DCE shared library (`libdce.sl`)
- DCE Core Client Software:
  - Allows you to configure and administer a minimal DCE client host and to run any DCE RPC applications.
  - Includes:
    - DCE daemon and control program
    - CDS client and administrative software
    - Security client and administrative software
    - Kerberos tools
    - Administrative tools for UUIDs, status codes, etc.
    - DCE configuration tools (`dce_config` and sam-based DCM)
    - Message catalogs for all DCE runtime software
    - DCE Audit Service daemon

- DTS Software:
  - Allows you to synchronize DCE hosts.
  - Includes:
    - DTS daemons and administrative software
    - DTS time provider examples

## Optional-Bundled Software

*For 10.20 and 10.10:*

This software is optionally installable. That is, it is included on the HP-UX 10.20 and 10.10 media and is available for installation, but is not included in any of the default HP-UX bundles.

The optional-bundled software includes:

- DFS Client Software (in a future release, DFS client software will be distributed at no charge in the Applications Release media)
- Supplemental DCE Administrative Software (Account Manager, Cell Monitor, CDSBrowser, HP Camera, `dceping`, `dceval`, and so on.)
- DCE Application Development Software
- Manpages and Online Help for bundled software

Consult your HP Sales Representative (SR) or Service Engineer (SE) for details.

For further DCE product information, including configuration instructions and information on how US users can order the free “DCE/9000 U.S. Libraries for HP-UX” product, see the following:

*On a 10.20 system:*

- `/opt/dce/newconfig/RelNotes/HPDCE1.5RelNotes.ps`
- `/opt/dce/newconfig/RelNotes/HPDCE1.5RelNotes.txt`

*On a 10.10 system:*

- `/opt/dce/newconfig/RelNotes/HPDCE1.4.1RelNotes.ps`
- `/opt/dce/newconfig/RelNotes/HPDCE1.4.1RelNotes.txt`

## 5-16 Other Operating System and Subsystem Changes

---

## Diagnostics

*For 10.10:*

- Support all new hardware that was supported as of 10.0.
- Support 10.10 Threads and Large File System features.
- Support both Series 700 and 800.
- Distributed as “Support Media” on both platforms.

The “Support Media” replaces the Support Tape of 9.X releases (“Media” means a tape, disk, or set of either). The 10.x Support Media consists of a tape, disk, or in some cases a kit containing the following:

- Offline Diagnostics
- Recovery Kernel
- Online Diagnostics
- RCL Tools (Predictive)

The Offline Diagnostics and Recovery Kernel are in the form of a LIF volume intended to be run from the tape or CD-ROM in the same way as the 9.x Support Tape. The Online Diagnostics and RCL Tools are in one Software Distributor (SD) bundle; use SD to load them onto your system.

## Recovery

The Offline Diagnostics include a utility, `copyutil`, that allows you to create a “failsafe” image of your root disk for the purposes of crash recovery.

---

**Caution**      `copyutil` is the only way to create a recovery system on 10.x. The 9.x Series 700 utility `mkrfs` is no longer supported.

---

For instructions on installing diagnostics, making a recovery kernel or recovering your system after a crash, see the *Support Media User’s Guide* (92453-90010), which is included in your media kit.

*For 10.0:*

- Support all new hardware that was supported as of 10.0.
- Support the 10.0 file system layout, I/O convergence, and the new LVM features.

- Support both Series 700 and 800.
- Distributed as “Support Media” on both platforms.

The “Support Media” replace the Support Tape of earlier releases. (“Media” is HP shorthand for a tape or disk, or set of either.) The 10.x Support Media consist of a tape, disk, or in some cases kit, containing the following:

- Offline Diagnostics
- Recovery Kernel
- Online Diagnostics
- RCL Tools (Predictive)

The Offline Diagnostics and Recovery Kernel are in the form of a LIF volume intended to be run from the tape or CD-ROM in the same way as the 9.x Support Tape. The Online Diagnostics and RCL Tools are in one Software Distributor (SD) bundle; use SD to load them onto your system.

## Recovery

The Offline Diagnostics include a utility, `copyutil`, that allows you to create a “failsafe” image of your root disk for the purposes of crash recovery.

---

**Caution**      `copyutil` is the only way to create a recovery system on 10.x. The 9.x Series 700 utility `mkrs` is no longer supported.

---

For instructions on making a recovery system, and recovering your system after a crash, see the *Support Media User’s Manual* (92453-90010), which is included in the Support Media kit.



---

## Dynamic Buffer Cache

The 9.01 Series 700 dynamic buffer cache functionality is provided on 10.x Series 800 computers. The programming interfaces have not been changed. The configuration parameters *nbuf* and *bufpages* specify buffer cache size. When these parameters are not specified on 9.0 Series 800, a default percentage of 10 percent of available memory is used. Not specifying these parameters in the 10.x kernel will configure a dynamic buffer cache as in 9.0 Series 700. If these parameters are specified in the 10.x kernel, they will indicate a fixed size buffer cache as in the 9.0 Series 800 release.

Using dynamic buffer cache causes the buffer cache to grow or shrink while the system is running (in response to competing demands for memory.) The actual portion of physical memory used in the buffer cache, expressed as a percentage, is bounded by the kernel configuration parameters *dbc\_min\_pct* and *dbc\_max\_pct*.

In many cases, file system I/O speeds up when there are no competing demands for memory. If the minimum and maximum limits are set to the same value, the size of the cache is fixed and behavior and performance should be the same as with 9.0 Series 800 systems.

On small- and medium-sized machines, file system access should be faster when paging traffic is low. There may be no speedup or slight degradation on very large systems. In such cases, create a fixed-size buffer cache by setting the *dbc\_min\_pct* and *dbc\_max\_pct* kernel configuration parameters to the same value to fix the size of the buffer cache. (For example, set them both to 12 for a buffer cache that uses exactly 12 percent of memory.)

---

## Fast Symlinks

HP-UX 10.x includes a *create\_fastlinks* kernel parameter that, when non-zero, causes HFS symbolic link creation to change so that one extra disk block access is avoided for each symbolic link in a pathname lookup. This involves a slight change in the HFS disk format, which makes these disks unusable on pre-10.0 Series 800 systems or on pre-9.0 Series 700 systems. For this reason, the default setting of *create\_fastlinks* is 0, which avoids creating the newer, faster format. All 10.x kernels (and all Series 700 9.0 kernels) understand both disk formats, however, whether *create\_fastlinks* is zero or non-zero. Use SAM to change the value of *create\_fastlinks*.

---

## File System Changes

### Rock Ridge

*For 10.10:*

HP-UX 10.10 includes the Portable File System (PFS) from Young Minds Inc. PFS allows mounting of Rock Ridge, ISO 9660 or High Sierra CD-ROM disks. ISO and High Sierra file names are all uppercase 8.3;1 format, so they look like

```
F00.EXE;1
```

Rock Ridge extensions to ISO 9660 allow full POSIX semantics and POSIX-like file names, so the file name

```
UFS_VNPS.C;1
```

becomes

```
ufs_vnodeops.c
```

on a Rock Ridge CD-ROM.

Even on ISO or High Sierra disks, PFS mount options allow:

## 5-20 Other Operating System and Subsystem Changes

- translation of file names from uppercase to lowercase.
- suppression of “;1” version numbers
- conversion of “;1” to “.1” suffix

CD-ROM disks mounted on a server machine may be exported to PFS clients.

Refer to the `pfs(4)` and `pfs_mount(1M)` manpages for more details.

Kernel-based support for CD-ROMS has also been extended to allow suppression of version numbers and lowercase matching and display of names. Refer to the `cdfs_mount(1M)` manpage and the description of the `cdcase` option.

## Journalled File System

On HP-UX 10.0 and earlier releases, the High Performance File System (HFS) is the standard file system and is the only locally mounted read/write file system available. With HP-UX 10.01, however, non-root file systems can use the Journalled File System (referred to as either VxFS or JFS). HP-UX’s implementation is based on the version from Veritas Software Inc.

VxFS is an extent-based journaling file system that offers fast file system recovery and online features such as online backup, online resizing, and online reorganization. It supports all existing file system interfaces except HP Access Control Lists (ACL) and B1 security features. VxFS cannot be a root or bootable file system in 10.01.

There are two VxFS products: base and advanced. The base VxFS file system has a fast recovery feature and is provided on all 10.01 systems. Advanced VxFS also includes all of the online features, and is available as an optional product.

HP-UX libraries and commands/utilities have been modified to support VxFS, and HP-UX commands have been updated to handle both HFS and VxFS as transparently as possible. You can continue to use all the commands you are used to, and they will continue to work with the HFS file system as they always have. These same commands also support VxFS, but will perhaps have different options that are meaningful to VxFS. Because all existing file system administration commands (such as `mkfs`, `newfs`, `fsck`, and `mount`) have been updated to support both HFS and VxFS, once a VxFS or HFS file

system is created and mounted through SAM, SAM will set up the file system such that regular maintenance of the file systems will continue in a manner similar to previous system administration. SAM has been updated to support basic VxFS, with the exception of online resizing, online backup, and online reorganization, which must be done from the command line.

Your existing applications will continue to work with HFS, and most of them will also work with VxFS. The only applications that will not work with VxFS are those that are aware of HFS-specific information, such as HFS disk format, read/write HFS file systems using the raw or character device, and so on. These applications cannot deal with any file system types other than HFS because of their dependency on the HFS disk format. All applications that use portable file system interfaces such as those specified by POSIX, X/Open, AES, SVID, and so on will work with all file system types that support these interfaces.

For an HP-UX file system overview and information on converting a non-root HFS file system to VxFS, refer to Chapter 4, “Working with HP-UX File Systems,” in the 10.01 version of the *HP-UX System Administration Tasks* manual.

## Multiple File System Types

The interface for specifying and detecting file system types has been expanded in 10.01 so that file systems from HP and third-parties will be handled more consistently.

In 9.0, there is a constant in `mount.h` for each file system type: `MOUNT_UFS`, `MOUNT_CDFS`, and `MOUNT_NFS`. These constants are used in `vfsmount(2)`, and are returned by `stat(2)`, `statfs(2)`, and `statfsdev(3)`.

To handle more file systems on HP-UX 10.01, an interface similar to AT&T's SVR4 is used to specify the file system type. The following is a summary of changes:

- `vfsmount(2)` is no longer supported, though existing calls still work. Use `mount(2)` instead.
- `mount(2)` is still supported in its old form (HFS only), but also has a new form in which the file system type is specified as a string.
- `stat(2)`, `statfs(2)`, and `statfsdev(3)` still return file system types as integers. For HFS, NFS, and CDFS, the mount constants in `mount.h` are still

## 5-22 Other Operating System and Subsystem Changes

valid. However, the more general method for translating types from integers to strings, or the reverse, is to use `sysfs(2)`, which works with both old and new file systems. Using `sysfs(2)` does the mapping at run time rather than compile time, which allows you to assign file system type numbers dynamically.

- The new routines `statvfs(2)` and `statvfsdev(3)` return the file system type as a string directly. The older routines `statfs(2)` and `statfsdev(3)` are still supported.

---

## Graphics

*For 10.20:*

The 3D graphics APIs (Starbase, HP PEX, and HP-PHIGS) for HP-UX 10.20 are all supersets of the versions supported on HP-UX 10.10.

Starbase, HP PEX, HP-PHIGS, and PowerShade now support X11R6 and two new graphics devices: HP VISUALIZE-EG and HP VISUALIZE-48XP. In addition, Starbase and HP PEX both have new entry points available. For information on what is new for 3D graphics at HP-UX 10.20, see the following on-line release notes:

*For HP PEX:*

```
/opt/graphics/PEX5/newconfig/opt/graphics/PEX5/10.0_Rel_Notes
```

*For HP-PHIGS:*

```
/opt/graphics/phigs/newconfig/opt/graphics/phigs/PHIGS.3.0.ReleaseNotes
```

*For Starbase:*

```
/opt/graphics/starbase/newconfig/opt/graphics/starbase/10.0_Rel_Notes
```

*For 10.10:*

The *HP PEX/PEXlib 5.1, version 3 (5.1v3) 3D Graphics Release Notes* is in the file `/opt/graphics/PEX5/newconfig/ReleaseNotes`. Use the `more` command to view this ASCII file or use this command to print the file:

```
lp /opt/graphics/PEX5/newconfig/ReleaseNotes
```

The *HP-PHIGS Information for Release 10.10 HP-UX* document is in the file `/opt/graphics/phigs/PHIGS.3.0.ReleaseNotes`.

The *Starbase Information for Release 10.10 HP-UX* document is in the file `/opt/graphics/starbase/10.0_Rel_Notes`.

The *Common Graphics Information for Release 10.10 HP-UX* document is in the file `/opt/graphics/common/10.0_Rel_Notes`.

*For 10.10:*

As of HP-UX 10.0, Starbase was no longer supported on the Series 800. However, it is still supported on the Series 700. In addition, three new graphics devices (VISUALIZE-8, VISUALIZE-24, and VISUALIZE-48) are available on Series 700 computers running the HP-UX 10.10 system. These devices are supported by the HP PEX, HP-PHIGS, and Starbase APIs.

HP-UX 10.x supports HP PEX 5.1, version 2, or “PEX 5.1v2” for short. HP PEX is an Application Programmer’s Interface (API) that brings high-performance, full-featured 3D graphics through the X server to the HP 9000 Series 700 workstations.

The location of `/usr/lib/libPEX5.sl` changed at 10.0. The new path is `/opt/graphics/PEX5/lib/libPEX5.1`. You may need to add the following statements to your makefile:

```
-L/opt/graphics/PEX5/lib/libPEX5.1
-I /opt/graphics/PEX5/include
```

---

## HP Process Resource Manager (HP PRM)

HP Process Resource Manager (HP PRM) is a new product that became available at HP-UX 10.0. HP PRM is a resource-management tool that is used to control the amount of CPU that processes can use. Basically, HP PRM is an alternate process scheduler that enables you to group processes together and guarantee each group a minimum share of the total CPU time. It consists of a configuration file and a command line interface to help you to create and manage workgroups of users, allocate CPU for each workgroup (and the

### 5-24 Other Operating System and Subsystem Changes

associated users), monitor the actual performance, dynamically change CPU allocation for workgroups, and move processes from one workgroup to another.

In addition, two of the core commands, `ps(1)` and `id(1)`, provide an option that displays the process resource manager ID (PRMID) information per process. HP PRM is integrated with HP GlancePlus (a real-time diagnostic monitoring tool) to display PRM CPU entitlements and to monitor the actual CPU usage by PRM group and processes within PRM groups.

For more information on HP PRM, refer to the *HP Process Resource Manager User's Guide* manual (B3834-90001).

The following is ordering information for HP PRM:

<b>Product Number</b>	<b>Description</b>
B3835AA	HP PRM License for HP9000/Series 800
B3834AA	HP PRM for HP9000/Series 800; Media and Manual
B3947AA	HP PRM License for HP9000/Series 700
B3948AA	HP PRM for HP9000/Series 700; Media and Manual

---

## **HP Remote Watch and HP Support Watch**

### **HP Remote Watch**

As of HP-UX 10.0, HP no longer supported HP Remote Watch.

HP Remote Watch will remain available and supported on all currently supported HP 9000 systems and workstations through 9.x, but it will not run on any new systems.

The functions of HP Remote Watch have been taken over by other tools that handle system management more effectively, such as the HP-UX System Administration Management (SAM) tool and HP OpenView OperationsCenter application.

## HP Support Watch

HP Support Watch on Series 800 systems used to include both HP Remote Watch and HP Predictive Support/UX; it will now consist only of Predictive Support/UX and will be renamed HP Predictive Support/UX.

## Who To Contact about HP Remote Watch and HP Support Watch

If you any questions, concerns or comments regarding HP Remote Watch or HP Support Watch, please contact Laurie Robell at Hewlett-Packard's Multivendor Services Division, via phone at (415) 691-5970, or e-mail/internet: laurie\_robell@hp5000.desk.hp.com.

---

## Instant Ignition

In the HP-UX 9.x releases, Instant Ignition provided functionality for system preconfiguration and a variety of enhancements to the user interface. Some of these features were available in the core HP-UX product. Others were part of the Ignition Plus ISU which was loaded with HP-UX on factory-installed Series 700 systems. These features included screens to help configure the system during initial boot-up, modules which summarized boot-time information into a brief checklist, localized welcome screens, productivity tools, a console window, and various HP VUE icons and actions.

Beginning with HP-UX 10.01, the Ignition Plus ISU has been obsoleted along with some of the features of Instant Ignition that were once part of core HP-UX. Whether you manually install 10.01, order 10.01 installed at the factory, or upgrade to 10.01 from 9.x, the following features will no longer be part of your system:

- UpdateFilesets, RemoveFilesets, and Updist icons/actions.

Update and rmfn functionality have been superseded by SD.

- HaltSystem, RebootSystem, and SingleUserMode icons/actions.

These three actions have been replaced by a single action, ShutdownSystem, which calls the shutdown dialog from SAM.

## 5-26 Other Operating System and Subsystem Changes



- System\_Info toolbox.

Information about the products installed on your system can now be found using the SD command `swlist`.

- Peripherals icon/action.

The peripherals on your system can be identified using SAM.

- Ignition script `configure.sh`.

At 9.x, you needed to run a script to make a manually installed system look like one that had been “ignited” at the factory. At 10.0, this script is not necessary because manually installed and factory-installed systems look identical (except for the system language variable on systems ignited with a non-English HP-UX bundle).

- Console window.

Although the console window is no longer provided as a default part of the HP VUE environment, one can be created easily using the Console command under the Utilities toolbox.

- Fun and Games directory.

The Fun and Games directory and all of the applications it contains (Flip, Hextris, Mines, XPilot, Xeyes, and ScrollBar) are not included in 10.01 and are removed when a 9.x system is upgraded to 10.01. However, the directory and any user-created action icons within the directory will remain, but be moved to the appropriate 10.01 location so they appear correctly in the upgraded HP VUE.

You can upgrade without losing the Fun and Games applications by following the steps shown below, but it is important that you remember that these applications are not supported on HP-UX 10.01 and have not been tested.

- Login as root and enter:

```
cat /etc/filesets/I+FUN-GAMES | cpio -ocv > /usr/contrib/fungames
```

- Perform the upgrade including transition links (default).

- As root, enter:

```
cpio -icdumvxl < /usr/contrib/fungames
```

- Execute the ReloadActions action under the System\_Admin toolbox.

The Series 700 Office Tools (Datebook, the `xhpcalc` calculator, the `xcal` calendar, and the `xdialog` utility) are included with 10.01, but they will *not* be supported in future releases. Although these tools will continue to run on future 10.x releases, they will not be included with the HP-UX product, nor will they be available elsewhere. These tools will be replaced by applications offering similar functionality.

---

## Kermit

The former version of kermit (C-Kermit, 4E(072) 24 Jan 89) was replaced by C-Kermit, 5A(190) 1993 at 10.0. The executable file on a 10.x system is `/usr/bin/kermit`.

C-Kermit 5A(190) is a full-function communications software package for HP-UX 10.x from Columbia University in the City of New York. It is furnished to you as part of HP-UX 10.x by special agreement between Hewlett-Packard Company and Columbia University.

Technical support for C-Kermit 5A is provided solely by Columbia University and not by Hewlett-Packard. To obtain support, call Columbia University at 1-212-854-5126, or send electronic mail to `kermit@columbia.edu`, or write to Kermit Distribution, Columbia University, 612 West 115th Street, New York, NY 10025-7221.

Prior to requesting technical support you need to have purchased and consulted the *Using C-Kermit* user manual, and you should also have consulted the `ckcker.bwr` and `ckuker.bwr` files in `/usr/share/lib/kermit`.

For more information, see:

- The README file `/usr/share/lib/kermit/READ.ME` on your 10.x system.
- The 10.x `kermit(1)` manpage.
- The following books, which you can get from bookstores, direct from the publisher, or from Columbia University:
  - *Using C-Kermit*, by Frank da Cruz and Christine M. Gianone, Digital Press Butterworth-Heinemann, Woburn, MA, USA (1993).

### 5-28 Other Operating System and Subsystem Changes

This is the C-Kermit user manual, tutorial and reference.

- *Kermit, A File Transfer Protocol*, by Frank da Cruz, Digital Press Butterworth-Heinemann, Woburn, MA, USA (1987).

This is the Kermit file transfer protocol specification.

To order Kermit books, call Columbia University at (212) 854-3703, or Butterworth-Heinemann at (800) 366-2665.

---

## Kernel Configuration

*For 10.10:*

The kernel configuration area now recognizes Fibre Channel drivers.

---

## Kernel Tunables

*For 10.10:*

The following default values have changed for system tunables. Applications that depend on the 9.x releases default values may need to adjust these parameters when building the 10.x kernel.

System Tunable	Default Value 9.x	Default Value 10.x
msgmax	32768	8192
msgmnb	32768	16384
msgseg	7168	2048
msgtql	256	40

### New Kernel Tunable (`o_sync_is_o_dsync`)

*For 10.10:*

The `o_sync_is_o_dsync` kernel tunable parameter is new for 10.10, but is available as a patch for 10.0 and 10.01.

`o_sync_is_o_dsync` provides a way for you to get 9.x synchronous IO semantics and performance on 10.x releases with 9.x binaries that use the

`O_SYNC` option to `open(2)` or `fcntl(2)`. In 9.x, `O_SYNC`, like `O_DSYNC` in 10.0, caused synchronous writes for file data, but not for file attributes that are not needed to retrieve the data, such as access and modification times. This gave improved performance, but was a violation of the POSIX standard. In 10.0, `O_SYNC` performed synchronous writes for data and all file attributes, while `O_DSYNC` is equivalent to 9.x `O_SYNC`. If you do not want to rewrite code or recompile your 9.x `O_SYNC` applications, use the `o_sync_is_o_dsync` kernel tunable. When `o_sync_is_o_dsync` is set to a positive value, all `open(2)` and `fcntl(2)` calls that specify `O_SYNC` are translated into `O_DSYNC` inside the kernel, making the 10.x kernel equivalent to the 9.x kernel in synchronous file IO behavior. See the `open(2)` manpage for more information about `O_SYNC` and `O_DSYNC`.

Patches for `o_sync_is_o_dsync` is available for 10.0 and 10.01. For 10.10, SAM can be used to set `o_sync_is_o_dsync`; for 10.0 and 10.01, `o_sync_is_o_dsync` will have to be set using `adb`.

This feature is off by default, so will have no effect on users who do not turn it on. When this option is turned on, all applications and processes running on the system that have files opened or `fcntl'd` with `O_SYNC` will have the `O_SYNC` changed to `O_DSYNC`. In the event of a kernel crash, changes to file attributes that are not needed to access file data may be lost, but file data will not be lost.

Use of this option preserves backward compatibility for performance purposes with 9.0X releases. If you do not use this option, applications that use `O_SYNC` in 9.0X and that do not modify their sources, might see a negative performance impact. But, functionally the applications will run the same.

Users who want to have only those file attributes necessary to retrieve user data written out synchronously and who do not want to use the `o_sync_is_o_dsync` tunable should recode their applications to use the `O_DSYNC` flag instead of the `O_SYNC` flag. See the `open(2)` and `fcntl(2)` manpages for details.

## 5-30 Other Operating System and Subsystem Changes

---

## MC/ServiceGuard

MC/ServiceGuard is a Series 800 high-availability product that was introduced at HP-UX 10.0. MC/ServiceGuard provides an environment in which, if a node fails, services (applications) can be up and running again on another node very quickly. Refer to the *Managing MC/ServiceGuard* manual (B3936-90001) for more information on the MC/ServiceGuard product.

Although MC/ServiceGuard includes functionality that is similar to that of Switchover/UX, the Switchover/UX product is also supported on HP-UX 10.x. If you are upgrading your SwitchOver/UX software from HP-UX 9.x to HP-UX 10.x, you must edit a set of configuration files after performing the transition to HP-UX 10.x. The process is explained in a separate document: *Read Me Before Upgrading SwitchOver/UX to HP-UX 10.x* (92668-90007).

---

## Measurement Systems (KI/MS)

Kernel Instrumentation (KI) is provided on both Series 700 and Series 800 computers for 10.x. In addition, the Measurement Systems (MS) interface has been removed from the kernel. A compatibility filter converts KI trace information to the format produced by the `ioscan(1M)` filter, so all programs depending on this interface will continue to work without modification. The `msmod()` program will also be modified to provide the same interface to measurement system configuration.

---

## Memory

This section describes the following memory topics:

- Virtual Memory
- Memory Page Deallocation (MPD)
- Memory Mapped Files and Semaphores

## Virtual Memory

The 10.x kernel includes the development work done for the 9.0 Series 700 to improve kernel responsiveness to memory pressure. The development work done for the 9.0 Series 800 to relieve swap space pressure is also included. Total virtual space is equal to pageable-RAM plus physical swap. This allows you to boot a system without physical swap (usually specified in the kernel system file) and add physical swap space at runtime through `/etc/fstab` or the `swapon(1M)` command.

## Virtual Memory Policies

Series 700 and Series 800 computers now use the same pager, and the swapper on both Series has been replaced by a deactivation scheme.

The following is a summary of major changes.

- A new `serialize()` system call is available which causes a process to run serially with other processes also marked with this command. If you have a number of large database applications or other batch-type applications, you can use the `serialize()` command to run these applications one after another instead of running them all at the same time; this dramatically decreases problems caused by processes competing for CPU and memory resources (sometimes referred to as “thrashing”). For more information on `serialize`, refer to the `serialize(1)` and `serialize(2)` manpages.
- The `nice()` system call influences how aggressively the pager will steal pages from a process. The “more nice” the process is, the more likely it is that the pager will steal its pages. The HP-UX 9.0 pager treats all processes equally, and the `nice()` command is used to modify process priority in relation to CPU usage. In 10.x, process priority can also affect how much of the process stays in memory, thereby improving system response time for processes that are not “niced.” For more information on `nice`, refer to the `nice(1)` manpage.
- In 10.x, processes are deactivated, not swapped. In 9.0, the entire process is pushed out to disk if it is swapped out. In 10.x, a process is deactivated (that is, it is taken off of the run queue) instead. The deactivated process’ pages are pushed out to disk over time by the pager.

## 5-32 Other Operating System and Subsystem Changes

- Process deactivation occurs when the system detects processes competing for CPU or memory resources as well as when the system experiences memory pressure. In 9.0, swapping is triggered only by low-memory conditions.

Since `sched()` is not always able to deactivate a process right away (for example, when the process is doing I/O), in 10.x, `sched()` has the ability to mark a process ready for deactivation. The next time the process faults, it deactivates itself. In 9.0, `sched()` just skips the process if it is busy, even if it is the best process to swap.

- To make memory more available for higher priority processes, only high-priority processes are able to drive free memory to zero before waiting for memory. Lower priority processes can sleep while waiting for memory before free memory reaches zero.

### **Memory Page Deallocation (MPD)**

HP-UX 10.x includes Memory Page Deallocation (MPD), a memory fault tolerance feature that allows memory boards to continue to function after a failure. MPD deallocates 4-KB pages and keeps track of them in the Page Deallocation Table (PDT). The PDT is maintained by PDC and contains entries for DBEs (double bit errors), hard SBEs (single bit errors that are reproducible, or two occurrences of the same SBE within a certain time period), the Golden Memory size (a segment of DBE-free memory for the kernel to reside in), the IS (interleave status) bit, and the CS (PDT clear status) bit. The table is stored in non-volatile RAM, so the contents are preserved between system boots.

The kernel and diagnostics support online deallocation of bad pages on both Series 700 and Series 800 systems. However, because the PDT requires firmware (PDC) support, only the following systems can deallocate the same bad pages between system boots: Series 800 Model T-500, Series 800 Model 890, and Series 800 Model K on future releases of 10.x.

## Memory Mapped Files and Semaphores

Memory Mapped Files and Semaphores are provided on both the Series 700 and Series 800 for 10.x. The following commands were new at 10.0 on the Series 800 and are unchanged on the Series 700:

```
fcntl(2)
fork(2)
truncate(2)
madvise(2)
mmap(2)
mprotect(2)
msem_init(2)
msem_lock(2)
msem_unlock(2)
msync(2)
munmap(2)
sysconf(2)
mman(5)
stat(5)
```

For more information on these commands, refer to their associated manpages.

The HP-UX 9.0 memory mapped file policies are simple: Up to 16 pages can be grabbed per I/O, even if you are accessing the file randomly. There is no characterization of the user access patterns nor any “read-ahead” for sequential access. However, HP-UX 10.x has enhanced pagein/pageout paths for both the NFS and HFS file systems. Attention is paid to user access on files. Those exhibiting random access (faults that are not sequential), do less I/O on each fault. Those processes doing reads and writes in sequential order have their I/O’s increased in size up to a maximum of 64 pages, including I/O’s launched asynchronously in anticipation of future access.

Refer to the `madvise(2)` and `mmap(2)` manpages for more details.



---

## Multimedia for 10.20

*For 10.20:*

This section covers the capabilities available from the Media Toolbox in the General Toolbox. On your Series 700 workstation or HP ENVIZEX or ENTRIA X station, you can use the Media Toolbox for:

- Capturing screen images.
- Viewing, editing, and printing images.
- Playing, recording, and editing audio.

### Capturing Screen Images

Capture allows you to take a picture of part of your display and save it as a file. This is also known as a “window dump.”

You can use two methods to capture images from the screen:

- The Capture control in the Media Toolbox. You can capture windows with or without borders, all of your screen, or part of your screen. You can save the screen image in one of several types of graphics files. Also, you can schedule the capture to occur immediately or select a future time for the image to be captured.
- The Capture Screen . . . button in the SharedPrint/UX dialog, if SharedPrint/UX is enabled on your system. This button presents the same choices for what you capture; however, you print the image without saving it to a file. See “SharedPrint/UX” for details on SharedPrint/UX.

### Viewing, Editing, and Printing Images

ImageView provides a quick way to view graphics, images, or screen captures in full color. Once the image is displayed, you can cut, copy, and paste sections, adjust the brightness and contrast, rotate the image, convert the file type, and save these changes.

While previously ImageView only displayed images, it now has the editing features of ImageView in HP MPower (an optional HP-UX 9.x product) plus Cut, Copy, and Paste features.

ImageView supports many types of image and graphics files, such as TIFF, JPEG, and XWD. Refer to the ImageView online help for a full list of file types.

For more information, see “Viewing Images” and “Printing an Image File”.

### **Viewing Images**

To edit an image, you must first display it.

1. Display the File Manager window that contains the image you want to see.

You can use sample image files in the `/opt/image/lib/help/C/graphics` directory. These image files appear in ImageView online help.

2. Double click the image file’s icon.

*Or*, drop the image file’s icon on the Image control in the Media Toolbox.

### **Printing an Image File**

To print images, you must have access to a SharedPrint/UX printer. For basic SharedPrint/UX information, see “SharedPrint/UX”.

For more information on SharedPrint/UX, refer to the *SharedPrint/UX User and Administrator’s Guide for HP-UX 10.0* (B1171-90124).

### **Playing, Recording, and Editing Audio**

On most Series 700 workstations and certain X stations, you can play audio or record it from a variety of sources, such as a tape player or CD player. You can send the signal to a variety of outputs, such as your headphones, a tape recorder, or external speakers.

Recordings are stored as audio files. An audio file can be played, or it can be displayed as a waveform. You can edit the waveform. Editing the waveform is similar to editing a text file; you can add to the audio waveform, delete a segment, and move or copy segments.

Creating an audio file is different from adding audio annotation to a file. Refer to the *HP VUE User’s Guide* for information on audio annotation.

## **5-36 Other Operating System and Subsystem Changes**

## Supported Workstations and X Stations

All HP 9000 Series workstations have digital audio hardware as standard equipment except for the Model 720, 730, and 750. In addition, HP ENVIZEX and HP ENTRIA X Stations equipped with an audio accessory kit can use the X station for playing audio via the HP Audio Subsystem.

## Setting Up Audio

If you are using a workstation, it needs to be on a network that is running HP DCE/9000 software. For more information, see the online DCE/9000 Release Notes in the following files:

```
/opt/dce/newconfig/RelNotes/HPDCE1.5RelNoteClient.ps  
/opt/dce/newconfig/RelNotes/HPDCE1.5RelNoteClient.txt
```

To set up a microphone, speakers, or other audio equipment, see the *Using Your HP Workstation* manual (A2615-90003) or the Audio Editor help (look under “Audio Editor Tasks”).

To connect audio equipment to an ENVIZEX or ENTRIA X station, see the owner’s manual for your X station.

## Using the Audio Control Panel

The Audio Control panel controls playback volume and determines where you hear the audio. It also provides a way to stop playback or monitor a recording.

On the VUE Front Panel, the Audio Control is to the right of the Text Editor control. On the CDE Front Panel, the Audio Control is to the right of the Mailer Control. Click the Audio Control to start it. If an error message appears with a Help button, click the Help button for instructions. You can use these instructions to troubleshoot the problem.

## Starting the Audio Editor

1. Display a File Manager window containing the file you want.

To use a sample audio file, use a file in the `/opt/audio/sounds` directory.

2. If you are using HP VUE, display the Media Toolbox.

Or, if you are using HP CDE, select the Digital\_Media application group in the Application Manager.

3. Drop the icon for the audio file onto the Audio control.

Audio files require a large amount of storage space. Keep only the files that you need and delete them when you are finished using them to avoid running out of disk space.

### Using Multimedia at the Command Line

To use the multimedia tools at the HP-UX command line, refer to the manpages. See Table 5-1 for a list of manpages.

**Table 5-1. HP MPower Manpages**

Component	Manpage	Topic
Capture	<code>capture(1)</code>	Capture a screen image to a file
ImageView	<code>imageview(1)</code>	Start up ImageView
SharedPrint/UX	<code>sprint(1)</code>	Start up SharedPrint/UX
	<code>spadmin(1M)</code>	Start the SharedPrint/UX manager
	<code>setup_pr.sh(1M)</code>	Configure printers for SharedPrint/UX
Audio	<code>audio(5)</code>	Start the Audio Editor or the Audio Control Panel
	<code>aserver(1M)</code>	Control the audio server (Aserver)
	<code>asecure(1M)</code>	Change audio permissions
	<code>attributes(1)</code>	List an audio file's attributes
	<code>convert(1)</code>	Convert an audio file
	<code>send_sound(1)</code>	Play an audio file

---

## Multimedia for 10.10

*For 10.10:*

This section covers the capabilities available from the Media Toolbox in the General Toolbox. On your Series 700 workstation or HP ENVIZEX or ENTRIA X station, you can use the Media Toolbox for:

- Capturing screen images.
- Viewing, editing, and printing images.
- Playing, recording, and editing audio.

### Capturing Screen Images

Capture allows you to take a picture of part of your display and save it as a file. This is also known as a “window dump.”

You can use two methods to capture images from the screen:

- The Capture control in the Media Toolbox. You can capture windows with or without borders, all of your screen, or part of your screen. You can save the screen image in one of several types of graphics files. Also, you can schedule the capture to occur immediately or select a future time for the image to be captured.
- The Capture Screen . . . button in the SharedPrint/UX dialog, if SharedPrint/UX is enabled on your system. This button presents the same choices for what you capture; however, you print the image without saving it to a file. See “SharedPrint/UX” for details on SharedPrint/UX.

### Viewing, Editing, and Printing Images

ImageView provides a quick way to view graphics, images, or screen captures in full color. Once the image is displayed, you can cut, copy, and paste sections, adjust the brightness and contrast, rotate the image, convert the file type, and save these changes.

While previously ImageView only displayed images, it now has the editing features of ImageView in HP MPower (an optional HP-UX 9.x product) plus Cut, Copy, and Paste features.

ImageView supports many types of image and graphics files, such as TIFF, JPEG, and XWD. Refer to the ImageView online help for a full list of file types.

For more information, see “Viewing Images” and “Printing an Image File”.

### **Viewing Images**

To edit an image, you must first display it.

1. Display the File Manager window that contains the image you want to see.

You can use sample image files in the `/opt/image/lib/help/C/graphics` directory. These image files appear in ImageView online help.

2. Double click the image file’s icon.

*Or*, drop the image file’s icon on the Image control in the Media Toolbox.

### **Printing an Image File**

To print images, you must have access to a SharedPrint/UX printer. For basic SharedPrint/UX information, see “SharedPrint/UX”.

For more information on SharedPrint/UX, refer to the *SharedPrint/UX User and Administrator’s Guide for HP-UX 10.0* (B1171-90124).

### **Playing, Recording, and Editing Audio**

On most Series 700 workstations and certain X stations, you can play audio or record it from a variety of sources, such as a tape player or CD player. You can send the signal to a variety of outputs, such as your headphones, a tape recorder, or external speakers.

Recordings are stored as audio files. An audio file can be played, or it can be displayed as a waveform. You can edit the waveform. Editing the waveform is similar to editing a text file; you can add to the audio waveform, delete a segment, and move or copy segments.

Creating an audio file is different from adding audio annotation to a file. Refer to the *HP VUE User’s Guide* for information on audio annotation.

## **5-40 Other Operating System and Subsystem Changes**

## Supported Workstations and X Stations

The Series 700 models 705, 710, 712, 715, 725, 735, and 755 have audio hardware. The models 720, 730, and 750 do not have audio hardware, but they can be upgraded to become models 725, 735, and 755. The audio on HP 9000 Series 400 workstations (which are not supported on HP-UX 10.x) is incompatible with the Audio Editor and Control Panel.

If you order HP ENVIZEX and HP ENTRIA X stations with an audio accessory kit, you can use the X station for audio.

## Setting Up Audio

If you are using a workstation, it needs to be on a network that is running HP DCE/9000 software. For more information, see the online DCE/9000 Release Notes in the following files:

```
/opt/dce/newconfig/RelNotes/HPDCE1.3RelNotes.ps  
/opt/dce/newconfig/RelNotes/HPDCE1.3RelNotes.txt
```

To set up a microphone, speakers, or other audio equipment, see the *Using Your HP Workstation* manual (A2615-90003) or the Audio Editor help (look under “Audio Editor Tasks”).

To connect audio equipment to an ENVIZEX or ENTRIA X station, see the owner’s manual for your X station.

## Using the Audio Control Panel

The Audio Control panel controls playback volume and determines where you hear the audio. It also provides a way to stop playback or monitor a recording.

In the Front Panel, the Audio Control is to the right of the Text Editor Control. Click this control to start it. If an error message appears with a Help button, click the Help button for instructions. You can use these instructions to troubleshoot the problem.

## Starting the Audio Editor

1. Display a File Manager window containing the file you want.

To use a sample audio file, use a file in the `/opt/audio/sounds` directory.

2. Display the Media Toolbox.

3. Drop the icon for the audio file onto the Audio control.

Audio files require a large amount of storage space. Keep only the files that you need and delete them when you are finished using them to avoid running out of disk space.

### **Changes from Audio on HP-UX 9.0**

If you used audio on HP-UX 9.0, note these changes to audio:

- Global volume control and line-out destination, which are accessible from the front panel using the same graphical user interface (GUI) that is used with HP MPower Version 2.x (an optional HP-UX 9.x product).
- Support for metering.
- A new audio editor that supports stereo and conversion of data and filetypes with file save. This is the same Audio Editor provided in MPower Version 2.x.

The Audio Developer's kit is no longer shipped with the HP-UX 10.x system. It is instead part of the HP-UX Developer's Toolkit (part number B3393AA for the Series 700, or part number B3395AA for the Series 800). This toolkit also includes the Image Developer's Kit and other programming capabilities.

### **Using Multimedia at the Command Line**

To use the multimedia tools at the HP-UX command line, refer to the manpages. See Table 5-2 for a list of manpages.



**Table 5-2. HP MPower Manpages**

Component	Manpage	Topic
Capture	<code>capture(1)</code>	Capture a screen image to a file
ImageView	<code>imageview(1)</code>	Start up ImageView
SharedPrint/UX	<code>sprint(1)</code>	Start up SharedPrint/UX
	<code>spadmin(1M)</code>	Start the SharedPrint/UX manager
	<code>setup_pr.sh(1M)</code>	Configure printers for SharedPrint/UX
Audio	<code>audio(5)</code>	Start the Audio Editor or the Audio Control Panel
	<code>aserver(1M)</code>	Control the audio server (Aserver)
	<code>asecure(1M)</code>	Change audio permissions
	<code>attributes(1)</code>	List an audio file's attributes
	<code>convert(1)</code>	Convert an audio file
	<code>send_sound(1)</code>	Play an audio file

---

## Peripheral Devices

*For 10.10:*

The following are now supported:

- Fibre Channel I/O cards
- 3490E 1/2" cartridge tape drive
- 8505XL 8mm tape drive
- Othello/Nestor DAT autochanger
- Quad-speed CDROM
- Ultralight Floppy

The only architecture-dependent change is the addition of C1300/C2300 disk array support on S700.

---

## POSIX Realtime Extensions

### Scheduling

HP-UX 10.x includes the following realtime features from the POSIX 1003.4 draft (D14) standard:

- Priority scheduling—A “realtime deterministic” scheduler (`rtsched(1)`) provides different execution scheduling policies. This scheduler co-exists with other HP-UX execution scheduling policies (timeshare, `rtprio`, and HP Process Resource Manager).
- Clocks and timers—The following, which co-exist with the existing HP-UX clock and (BSD) interval timer interfaces, have been added:
  - A standard set of interfaces to the existing clocks (time-of-day, process virtual, and process profiling).
  - A standard set of interfaces for creating and manipulating interval timers based on those clocks.
  - A high-resolution “sleep” function independent of interval timers and signals.
- A new configurable parameter, `rtsched_numpri`, controls the number of scheduling priorities supported by the new POSIX execution scheduler. The range of valid values is 32 to 512, and the default value is 32. Increasing `rtsched_numpri` provides more scheduling priorities at the cost of increased context switch time, and to a minor degree, increased memory consumption.

The new scheduler provides execution scheduling at the *highest* priorities. A process scheduled with it will be “stronger” than one scheduled at `rtprio 0`. This constraint is driven by the deterministic policy defined for the new scheduler. You may want to change applications that currently use `rtprio 0` to use the new scheduler if you want them to have priority.

## **ptrace() Enhancements**

The following new requests are included in the 10.x `ptrace()` system call. These requests allow you to debug processes with more control. For more details, refer to the `ptrace(2)` manpage.

- `PT_SET_EVENT_MASK`—used by the debugger to specify event information to a traced process.
- `PT_GET_EVENT_MASK`—used by the debugger to retrieve event information from a traced process.
- `PT_GET_PROCESS_STATE`—used by the debugger to retrieve state information logged by a traced process.
- `PT_GET_PROCESS_PATHNAME`—used by the debugger to retrieve the pathname string of an executable file exec'd by a traced process.

## **Synchronized I/O**

Synchronized I/O from the POSIX.4 standard is included in HP-UX 10.x with the following new features:

- Two new flags, `O_DSYNC` and `O_RSYNC`, which have been added for `open(2)` and `fcntl(2)`. `O_DSYNC` is a subset of the existing `O_SYNC` and allows `write()` to refrain from updating non-essential attributes synchronously and shorter execution time paths. `O_RSYNC` enables `read(2)` to fetch data afresh from the disk, synchronously writing out relevant delayed writes before returning from the read and providing a fresh-off-the-disk guarantee for applications.
- A new system call, `fdatasync()`, which is a subset of the existing `fsync(2)`, and which allows applications to refrain from updating non-essential attributes synchronously (independent of open flags). This allows shorter execution time paths.
- A new system call, `sysconf(2)`, announces the availability of synchronized I/O on the 10.x system. `pathconf(2)` and `fpathconf()` determine the availability of synchronized I/O for files in the HFS, VxFS, NFS, CDFS, and LOFS file systems.

---

## Programming Languages

This section summarizes changes that affect programming languages.

### libM Convergence

*For 10.20:*

At 10.20, there is a single C math library called `libm.a`, rather than separate SVID (`libm`) and XPG4 (`libM`) libraries. The single library is XPG4.2-compliant. Links are provided from each `libM` to the corresponding `libm` so if you call the math library, you do not have to change their makefiles. Compiling with `-lM` and `-lm` will both continue to work. Because the SVID and POSIX libraries were essentially the same at 10.0 (except for `matherr`), you should see no effect.

There are still `pa1.0` and `pa1.1` versions of the C math library. To support high-performing math code at `pa2.0`, modify the millicode library, as discussed below.

### matherr

*For 10.20:*

At 10.0, if a math routine is called with bad arguments, the library code generally invoked a user-written error handler called either `matherr` or `_matherr`, depending on the library. This behavior is gradually being obsoleted.

At 10.20, the two math libraries are converging and their common math error behavior supports *both* of the previous conventions. Both `_matherr()` and `matherr` will be invoked if a math routine is being called with bad arguments. If `_matherr` returns a non-zero value, `matherr` will not be called. At 10.20, both archive and shared math libraries support this combined `matherr` behavior.

In a future HP-UX release, the math library functions will no longer call `matherr` or `_matherr` when there is an error. Instead, there will be forward compatibility of shared-linked executables by versioning shared libraries. `/usr/lib/libm.2` will be added with the new `matherr` behavior while continuing to supply a `/usr/lib/libm.1` with the old 10.0 behavior. At this

## 5-46 Other Operating System and Subsystem Changes

future release, you will have to modify your applications so they do not depend on `matherr`. See the *Floating-Point Guide* for a sample wrapper function.

## **HUGE\_VAL**

*For 10.20:*

`HUGE_VAL`, which is currently defined in `<math.h>` as the largest double-precision value, will change to IEEE infinity at a future HP-UX release. This will coincide with a change in `libc`, which returns `HUGE_VAL` from certain string-to-float conversion routines.

## **Library Convergence**

*For 10.20:*

At a future HP-UX release, a separate `pa1.0` set of math libraries will not be shipped.

## **Millicode**

*For 10.20:*

New math entry points have been added to the millicode library (`/usr/lib/milli.a`). Because the compiler generates millicode, you can choose the name of these new entry points. For example, you can use the name `log_20` even though there is an existing `log` entry. The new routines are PA2.0 code and the compiler will only generate calls to the routines if `+DA2.0` appears on the compile line.

Because of the `+0libcalls` switch, which enables the generation of millicode calls for math routines, you can compile your programs and link them with 2.0-specific math routines. This allows you to obtain the full benefits of the pa2.0 architecture improvements in your floating-point applications.

## Functions Moved From libc to libm

*For 10.20:*

The functions `frexp`, `ldexp`, and `modf`, which have been supplied in HP-UX as part of the C library, are being shipped in the math library. This change has been made for XPG4.2 compliance. The functions will also remain in the C library for backward compatibility until the next release of `libc`.

## Automatic Vector Support in Math Library

*For 10.20:*

When programs are optimized with `+Ovectorize`, they can access some special vector support in `/usr/lib/pa1.1/libm.a`. Other than setting the optimizer switches, you do not have to make changes to obtain the resulting support routines. Among the routines provided are those that can move a vector, multiply by a constant, or take the sum of the elements. The High-Level Optimizer recognizes suitable loops in the user program and replaces them by math library calls. Programs compiled with `+DA1.1` or `+DA2.0` can benefit from this. Some of the routines in `/usr/lib/pa1.1/libm.a` are PA2.0 code, but the compiler ensures that they are only linked with 2.0 programs.

## Linker and Dynamic Loader

*For 10.20:*

### Linker

The linker (`ld`) and the dynamic loader (`dld.sl`) issue warning messages for any compatibility issues. These issues are described below. The messages, issued by default, can be turned off with the new option `+vcompatwarnings` and turned off with the option `+vnocompatwarnings`.

The following will issue a message:

- Linking any PA 2.0 object files. These object files, when linked, will not execute on a PA 1.x machine.
- The `-C` option (procedure call parameter and return type checking)
- The `-A` option (incremental loading)

## 5-48 Other Operating System and Subsystem Changes

- Any -H or -F options
- Linking object files that contain two symbols with the same name, but are of incompatible types. The most common situation would be a variable and a function that have the same name, but are in different object files.
- Library Created Storage (LCS) under certain circumstances. These messages only appear if the `v` option is also given. LCS can be detected by the following message:

`Loading <object file name>:<symbol name>`

A warning is issued only when LCS is used on an object file, but no functions within that object were used. This is because the object file is not checked for any unsatisfied symbols in the current toolset.

- Whenever a shared library is being created which uses Intra-library versioning.

### **Dynamic Loader**

The dynamic loader (`dld.sl`) also issues warning messages. By default, these message are off. The messages can be turned on by setting the environment variable `_HP_DLD_OPTS` to the include the string `warnings`. For example,

```
export _HP_DLD_OPTS=-warnings
```

One of the following can trigger a message:

- Certain routines and flags in the `shl_load(3x)` API. They include:

<code>shl_definesym()</code>	<code>shl_get()</code>
<code>shl_get_r()</code>	<code>shl_gethandle()</code>
<code>shl_gethandle_r()</code>	<code>shl_getsymbols()</code>
<code>BIND_FIRST</code>	<code>BIND_NOSTART</code>
<code>BIND_NONFATAL</code>	<code>BIND_VERBOSE</code>
<code>DYNAMIC_PATH</code>	<code>TYPE_PROCEDURE</code>
<code>TYPE_DATA</code>	<code>TYPE_STORAGE</code>

These may not be supported in a future 64-bit environment.

- Linking object files that contain two symbols with the same name, but are of incompatible types. The most common situation would be a variable and a function that have the same name, but are in different object files.

### **Impact**

The changes to the linker will result in more warning messages being issued. This might cause test suites and other programs which rely on the linker's output to fail.

The dynamic loader might cause a program to have a slower start-up time because the loader needs to check the environment before the program can be executed.

### **Performance**

The additional checking the linker must do most likely will have no impact on performance.

Program startup time might be affected because the dynamic loader checks the environment before each program is being executed.

### **Compatibility**

Programs that rely on the output of the linker might fail due to the extra messages.

### **Alternatives**

The LD\_OPTS environment variable can include the `+vnocompatwarnings` option instead of placing it on the command line. Similarly, the C and C++ compilers can be called with the `-Wl,+vnocompatwarnings` option to turn off the messages via the compiler.

### **Obsolescence**

The following features of the linker/dynamic loader might become obsoleted in a future HP-UX release (some only for a 64-bit environment):

- Procedure call parameter and return type checking (linker `-C` option)
- Incremental loading (linker `-A` option)
- Linker `-H` and `-F` options

## **5-50 Other Operating System and Subsystem Changes**



- Two symbols with the same name but incompatible types
- Library created storage
- Intra-library versioning
- Parts of the `shl_load(3x)` API (in a 64-bit environment)

## Fastbind

*For 10.20:*

The fastbind is a new tool that can improve the start-up time of programs that use shared libraries (incomplete executables) by storing information about needed shared library symbols in the executable file.

The following has been modified:

- The dynamic loader (`/usr/lib/dld.sl`) has been modified to emit symbol binding information to let the fastbind tool know the symbols used to bind executable. The `dld.sl` emits symbol binding information only when it is invoked by the fastbind tool.
- The fastbind tool (`fastbind`) will perform analysis on the symbols used to bind an executable and store this information in the executable file.
- The dynamic loader (`/usr/lib/dld.sl`) has been modified to process the fastbind information (in the executable) during program start-up and use this information to bind the executable instead of the standard search method for binding the symbols.
- The linker (`ld`) has been modified to accept the command line option `+fb`. When the `+fb` option is used, the linker will automatically run the fastbind tool on the incomplete executable it has produced.

## Large UID Support in the Archiver (AR)

*For 10.20:*

The archiver (AR) stores the user IDs (UID) and group IDs (GID) of the owner of the SOM file in the `ar_uid` and `ar_gid` members of the archive header `ar_hdr` (defined in the header file `/usr/include/ar.h`). The `ar_uid` and `ar_gid` used to be a decimal number (six ASCII characters), left-justified and blank padded. This resulted in a limit of 99999 on UID and GID. To support user and group IDs larger than 99999, the AR now uses a different

encoding for user and group ID values. The new encoding will support all values of the 32-bit signed integer UID and GID.

The following has been modified:

- The AR's new encoding will encode the 32-bit signed integer UID and GID using printable characters. It will encode six bits in each of the first five characters and two bits in the last character. The new encoding also ensures that the last character is not a blank. Old encoding will be used for numbers up to 99999.
- The `ar` has also been modified to detect the old UID and GID encoding based on the last character in `ar_uid` and `ar_gid` fields. The archiver will correctly decode the old and new encoding of UID and GID values.

## Large Files

*For 10.20:*

The following has been modified:

- The linker, dynamic loader, and the SOM tools will issue an error message when an input file is greater than 2GB in size.
- The `size`, `nm`, `chatr`, `strip`, and `odump` commands issue a diagnostic message when the input files are greater than 2GB (a large file):
- The system calls `open`, `lstat`, `fstat`, and `stat` will return `E_OVERFLOW` if used on a large file. These commands check for the error return value `E_OVERFLOW` and issue an appropriate diagnostic message.
- The dynamic load `/usr/lib/dld.sl` issues a diagnostic message if it encounters a shared library larger than 2GB.
- The linker (`ld`) and the archiver (`ar`) issue a diagnostic message when an attempt is made to create an output file larger than 2GB or if any of the input files are greater than 2GB.

## Linker On-line Help

*For 10.20:*

The linker (`ld`) supports a command line option `+help` to invoke the *HP-UX Linker and Libraries Online User Guide* if you are running an X window

## 5-52 Other Operating System and Subsystem Changes

system and the environment variable DISPLAY is set to the name of your workstation or X terminal.

### **Impact**

For UID and GID values greater than 99999, the `ar_uid` and `ar_gid` fields of `ar_hdr` contain the ASCII encoding of UID and GID and not the ASCII decimal number itself as it did previously.

### **Performance**

The start-up time of programs that have fastbind information might be less because the dynamic loader will use the fastbind information to bind the executable instead of the standard search method for binding the symbols.

### **Compatibility**

Programs that rely on the output of the linker might fail because of the extra diagnostic messages.

## **HP-UX Linker (ld)**

*For 10.20:*

The `-B symbolic` and `+0staticprediction` options have been added.

**Syntax:**

```
ld -b -B symbolic -o shlib1.sl obj1.o obj2.o ...
```

```
ld +0staticprediction obj1.o obj2.o ...
```

The `-B symbolic` option is used only in building shared libraries to direct the linker to resolve internal calls instead of building import stubs for those calls. Without specifying this option, the default is that unresolved procedures can be resolved either inside or outside the shared library; thus, import stubs are always built.

The `+0staticprediction` option sets the branch prediction bit in the output executable file's auxiliary header. This option is only applicable on the PA 2.0 architecture.

## Impact

Using the `-B symbolic` option in building shared libraries will limit library users from supplying their own customized procedures because all procedures, if defined inside the shared libraries, have already been resolved.

Using the `+0staticprediction` option, the bit in the executable file's auxiliary header will enable the branch prediction feature in the PA2.0 architecture. There should be no impact to your applications.

## Performance

For the `-B symbolic` option, there will be a slight performance improvement because the path length for internal calls will be shorter due to the elimination of import stubs.

For the `+0staticprediction` option, the branch prediction feature in the PA2.0 architecture should also improve performance.

## HP C/ANSI C

*For 10.20:*

Features:

- Numerous performance improvements, options, and support for the PA8000 architecture (see Performance section below).
- Addition of pragmas to support fast shared library calls (see Performance section below).
- Change in the default architecture for code generation and instruction scheduling (see Summary section below).
- A new option `-help` is supported, which opens a VUE help window for C on-line reference manual.
- Large file awareness supports input files larger than 2 GB bytes.

The defaults architecture for code generation and scheduling is as follows:

- PA2.0 machines generate code that runs on PA2.0 only.
- PA1.1 machines generate code that runs on PA1.1 and PA2.0. This code is optimized for PA1.1.

## 5-54 Other Operating System and Subsystem Changes

- PA1.0 machines generate code that runs on PA1.0, PA1.1 and PA2.0. This code is optimized for PA1.1.

In prior releases, the S800 compiler always generated PA1.0 code and the S700 compiler always generated PA1.1.

### Performance

There is additional performance tuning for the new PA8000 architecture, which is also known as the PA 2.0 architecture. The compiler supports `+DA2.0` and `+DS2.0` switches to generate peak performance for this architecture. Recompile for PA2.0 will show significant performance improvements. See the file `/usr/lib/sched.models` to see the mapping between models and PA architecture versions.

There are new pragmas for faster calls to shared libraries and improved performance. Refer to the *C Programmer's Guide* and this document for descriptions of these pragmas.

There are enhancements to optimization levels 2, 3, and 4 to enhance performance of programs, including macro definitions of `ABS`, `MIN`, and `MAX`.

The compiler continues to support the existing optimization switches. These switches are:

- `+0name ...` —detailed switches for individual controls
- `+0[no]data_prefetch`— to generate data prefetch instructions for data structures referenced within innermost loops.
- `+0entrysched`— to perform instruction scheduling on a subprogram's entry and exit sequences.
- `+0[no]fail_safe`— Fail safe optimization causes the compiler to automatically pick a lower optimization level instead of aborting the optimizer.
- `+0[no]loop_unroll[=unroll_factor]`— to control unrolling of loops.
- `+0fastaccess`—to enable fast access to global data items.
- `+0global_ptrs_unique`— to tell the optimizer whether there are unique global pointer variable names, and what the names are.
- `+0inline_budget`— to perform more aggressive inlining.

- `+0ptrs_ansi`— to assume that an `int *p` points to an `int` field of a struct/union, and that `char *` points to any type of object.
- `+0ptrs_to_globals`— to tell the optimizer whether global variables are modified through pointers.

### Obsolescence

In a future HP-UX release, the bundled C compiler (not the HP C/ANSI C compiler) will either be renamed or some current warnings will be changed to errors.

### HP C/HP-UX

*For 10.10:*

- The shared libraries `libl.1`, `libl.sl`, `liby.1`, and `liby.sl`, which were distributed in System Release 10.01 in the directory `opt/langtools/lib`, have been moved to the system directory `usr/lib`.
- The `cc` driver and the C compiler tools have been changed to be in conformance with the XPG4 standard.

In the principal change, the argument `--` now delimits the end of options. Any following arguments are treated as operands (typically, input file names), even if they begin with the `-` character.

For further information about enhancements made to HP C/HP-UX in System Releases 10.00, 10.01, and 10.10, read *HP C/HP-UX Release Notes Version A.10.3055* (5964-1375). This is also available online in the file `/opt/ansic/newconfig/RelNotes/ansic.10.30`.

For details of the changes made to the HP C/HP-UX compiler in System Releases 10.00 and 10.01, read *HP C/HP-UX Technical Addendum* (5963-4468).

For revised information on alignment modes and their use, read *HP C/HP-UX Release Notes Version A.10.01* (online only) located in the file `/opt/ansic/newconfig/RelNotes/ansic.10.01`.

*For 10.0:*

The HP C compiler product version A.10.00 release includes enhancements to the compiler, optimizer, and online and hardcopy documentation. The

## 5-56 Other Operating System and Subsystem Changes

following enhancements have been added to the HP C Series 700/800 compiler product since the A.09.61 release:

- New location for files.  
Makes it easier to administer HP C/HP-UX.
- New compiler option **+k**.  
Supports a change in how shared library global data is accessed by the program file.
- New compiler option **-V**.  
Causes each invoked subprocess to print its version information to **stderr**.
- Four-byte EUC support in filenames, strings and literals.  
Supports character sets that require four bytes to store each character.
- New millicode versions of math library functions.  
Improves performance. See “Math Library Routines (libm, libM)” in Chapter 7 for more information.
- Debugging of optimized code.  
Allows you to debug code that is optimized at level 1 or 2. The C compiler now provides support for debugging optimized code in conjunction with the HP Distributed Debugging Environment (DDE). This support is not provided in conjunction with **xdb**.
- New option **+ESnoparmreloc**.  
Disables parameter relocation for function calls.
- New optimization options **+0[no]procelim** and **+0[no]vectorize**.  
Provides finer optimization control.
- New **lint** functionality.  
Provides recognition for the HP C **-Wp** option, and provides an option that is equivalent to a **/\*LINTLIBRARY\*/** comment in source files.
- New diagnostic message information.  
Simplifies error recovery.

- Revised information describing alignment modes and their use.

Supplements the information in the HP C manuals.

For more information see:

- HP C/HP-UX Version A.10.00 Release Notes (5962-7312)
- HP C/HP-UX Technical Addendum (5962-7311)

## HP C++

*For 10.10:*

The new features available with HP C++/HP-UX Version A.03.72 are:

- New compiler option `+Xehdtcount`—to generate instrumentation that can locate run-time exception handling problems which cause run-time range errors.
- New syntax for the CXXOPTS Environment Variable—to allow options to be placed before and after the command line parameters to CC.
- Default parameter initialization—for class objects.

For more information, see your *HP C++/HP-UX Version A.10.09 Release Notes* and *HP C++ Troubleshooting Notes*. The release notes are available in the C++ product in hardcopy or in online form in `/opt/CC/newconfig/RelNotes/CXX.10.01`. The troubleshooting notes are available in the C++ product in `/opt/newconfig/RelNotes/tools.ps` or in `/opt/newconfig/RelNotes/ascii`.

*For 10.0:*

Changes for the HP C++ version, A.03.60, included:

- Online help and documentation changes.
- New locations for files and libraries.
- New compiler option for accessing shared library global data.
- New compiler option for using millicode routines. (See “Math Library Routines (libm, libM)” in Chapter 7).
- New compiler option to eliminate dead procedure code.
- New compiler option for nested shared libraries that contain templates.
- Support for multi-threaded applications (see “libc Changes to Support Threads” in Chapter 7).

## 5-58 Other Operating System and Subsystem Changes



- Four-byte Extended Unix Code (EUC) support.
- Debugging of optimized code.
- ANSI C compiler usage.
- Support for exception handling and shared libraries.

The HP C++ run-time libraries libC.ansi and libC are now shipped as part of the HP-UX core system. Prior to the HP-UX 10.01 release, these libraries were shipped with the HP C++ product.

The following files have been added to the core system at 10.01:

```
OS-Core:CORE-SHLIBS: /usr/lib/libC.ansi.1
OS-Core:CORE-SHLIBS: /usr/lib/libC.ansi.sl
OS-Core:CORE-SHLIBS: /usr/lib/libC.1
OS-Core:CORE-SHLIBS: /usr/lib/libC.sl

ProgSupport:LANG-MIN: /usr/lib/libC.ansi.a
ProgSupport:LANG-MIN: /usr/lib/libC.a
```

For more information, see the *HP C++/HP-UX Version A.10.00 Release Notes for HP 9000 Series 700/800 Computers*. These are on your 10.x system in the ASCII file `/opt/CC/newconfig/RelNotes/CXX.10.0`.

### **HP Micro Focus COBOL/UX**

HP Micro Focus COBOL/UX must be version B.09.10 or later to run on HP-UX 10.x.

For more information on HP Micro Focus COBOL/UX B.09.10 see:

- *HP Micro Focus COBOL/UX Version B.09.10 Release Notes* (5963-4420).
- *HP Micro Focus COBOL/UX Programmer's Guide, E0195* (B2433-90020).

### **HP FORTRAN/9000**

The 10.0 release of the HP FORTRAN/9000 compiler includes the following new features:

- Debugging optimized code.
- New optimization options.
- New `+k` compiler option.

- New format descriptors.

For more information about this product, refer to the *HP FORTRAN/9000 10.0 Release Notes*.

## **HP Pascal/HP-UX Release 10.0**

New features include:

- New optimizer options `+0[no]libcalls` and `+0[no]procelim`.
- Millicode versions of some built-in functions.
- Information on memory consumption when compiling at optimization level 4.
- Debugging optimized code (DOC).
- Thread-safe HP Pascal/HP-UX routines.
- Referencing shared-library data (`+k`).
- Four-byte extended UNIX code (EUC).
- Compliance with HP-UX 10.0 file system layout.
- Distributed Debugging Environment (DDE).
- New warning messages.
- Information on porting HP Pascal/HP-UX programs.

For more information see:

- *HP Pascal/HP-UX Release Notes* (B2415-90002).
- *HP Pascal/HP-UX Reference Manual* (92431-90006); not revised for 10.0.
- *HP Pascal/HP-UX Programmer's Guide* (92431-90005); not revised for 10.0.
- 10.0 `pc(1)` manpage for the Pascal compiler.

## **PA-RISC Assembler Release 10.0**

New features of the Precision Architecture RISC (PA-RISC) Assembler for Release 10.0 include:

- Additional characters in symbol names.
- Register type-checking in operands.
- Five new field selectors to handle three-instruction fixup sequences.
- New `.ALLOW` and `.LEVEL` directives, replacing the `+DA` and `+DS` compiler options.
- Additional keywords for `.CALLINFO` to support stack unwind descriptors.
- Additional symbol types supported for the `.IMPORT` directive.
- New `SHLIB_VERSION` directive.

## **5-60 Other Operating System and Subsystem Changes**

- Complete support for PA-RISC 1.1 features.
- Compliance with the HP-UX 10.0 file system layout.
- New and revised warning and error messages.

For more information see:

- *Assembly Language Reference Manual* (92432-90001).
- *Assembly Language Release Notes* (92432-90007).
- *PA-RISC 1.1 Architecture and Instruction Set Reference Manual* (09740-90039).

---

## **SAM (System Administration Manager) for 10.20**

### **On-line Help Localization**

On-Line Help for the Functional Area Launcher, Disks and Filesystems, and Users and Groups, will be localized into `ja_JP.SJIS` and `ja_JP.eucJP` locales.

### **Functional Area Launcher**

The Functional Area Launcher has been modified to handle a large number of users when the Add Custom Application or Modify Custom Application action is chosen. The dialogs associated with these actions allow the superuser to select a specific user to use when executing the custom application. A choices text edit is used to select the user on systems that have a small number of users, while a plain text edit is used for entry of a single user on systems with a large number of users.

### **Restricted SAM Builder**

The Load Privileges and Save Privileges task dialogs have been changed to display a text edit for entry of a single user or group when the Restricted SAM Builder is being run on a system that has a large number of users or groups. A list of users or groups from which a selection can be made is still displayed on these dialogs when the system being used has a small number of users and groups.

## **Trusted Systems**

User filtering is required prior to displaying the list of Audited Users when SAM is being run on a system that has a large number of users. A task dialog allows the administrator to specify a single user or a range of users to display on the Audited Users list.

To view the audit log, the administrator can filter on users. On systems with a small number of users, a list of users is displayed from which one or more can be chosen to filter on. Systems that have a large number of users will display an add-on list which shows the users currently selected for filtering on and which allows additional users to be added by typing in their login name.

## **Backup and Recovery**

Backup and Recovery will now support backing up a JFS file system in the same way as an HFS file system.

For a snapshot of a JFS file system, you must create the snapshot and then back the file system up by using SAM Backup & Recovery.

## **Users and Groups**

The User/Group area of SAM has been modified to support larger user and group IDs. User IDs and group IDs can now be as large as 2147483646. Changes have also been made in how large numbers of users or groups on a system are handled in SAM. If there are more than 500 users or groups defined on a system, SAM will ask you to select a subset of items to be displayed rather than attempting to display all of the users or groups in a list simultaneously.

## **Printers and Plotters**

The “Printers and Plotters” area of SAM now contains two primary subareas. One area provides HP-UX LP spooler management functionality (this area contains all of the functionality you are used to finding in the “Printer and Plotters” area of SAM). The other area provides HP Distributed Print Service (HPDPS) management functionality.

## **5-62 Other Operating System and Subsystem Changes**

The HP Distributed Print Service is a new print spooler that is provided in the 10.20 release of HP-UX. This spooler contains a much richer set of functionality compared to the LP spooler; is a true, distributed application and is based on the latest industry standards for printing. Navigate to the HPDPS subarea in SAM and select “What Is HPDPS?” to see a brief tutorial about this new print spooler and the management functionality provided in SAM.

### **Terminals and Modems**

A new option was added to terminals and modems that gives you the ability to modify a terminal or modem. This option brings up a task dialog similar to the Add Terminal or Modem action, giving you the ability to change the preconfigured parameters and save those changes.

When modifying authorized users on a trusted system with a large number of users (1000), an add-on list with an area to enter the user’s name will appear rather than the regular transfer list. This modification was made for performance reasons.

### **Performance Monitors**

This is a relocated area of SAM that used to be in the Process Management area and is now in the top level of SAM. This area also contains one new item called System Properties.

### **System Properties**

is a new task dialog in SAM that contains general system information split up into several groupings: Processor, Memory, Operating System, and Network. There is one additional area that shows dynamic information from the other areas. This area has an auto-refresh option.

### **Disks and File Systems**

- Users can create Logical Volume Manager (LVM) Physical Volume Groups (PVGs).
- Devices are now supported in a Fiber Channel environment.
- Version 2 VxFS file systems can now be upgraded to Version 3.

- The 4x library autochangers and disks are supported.
- The VxFS mount options are offered in four sets of useful combinations.
- SAM is large-file “smart”.
- SAM supports single-point administration of Shared LVM (SLVM) on high availability clusters.

## **UUCP**

UUCP configuration is no longer supported by SAM.

None of the changes described above are expected to affect current 10.10 SAM users.

---

## **SAM (System Administration Manager) for 10.10**

- Users can explicitly mark disks and/or logical volumes that are used for applications (such as databases) so that SAM does not consider these devices as unused.
- SAM supports striped logical volumes. This includes the ability to specify which disks are used when extending a stripe logical volume.
- SAM supports large file systems (greater than 4 Gbytes).
- SAM supports importing, exporting, activating and deactivating volume groups, with special extensions to these operations in Service Guard cluster environments.
- SAM supports the C1300/C2300 disk array on S700's. This support is done through the SCSI interface and does not require RS-232 communication links.
- The User/Group area of SAM has been modified to support systems where NIS has been configured. Users and groups can now be added, modified, and removed on systems running NIS. If SAM is run on an NIS client or a slave server, changes can be made to users and groups in the local `/etc/passwd` and `/etc/group` files only. If SAM is run on an NIS master server, user and group changes can be made and propagated to the NIS domain by rebuilding the NIS maps. In addition, SAM also supports having a separate password file on the NIS master server to build the NIS maps.

### **5-64 Other Operating System and Subsystem Changes**

- Using SAM, you can configure the DNS resolver and set up a local DNS name server. For more information, see the SAM online help.
- Using SAM, you can configure the local host as a Network Information Service (NIS) client, NIS master server, or NIS slave server. For more information, see the SAM online help.
- Using SAM, you can configure the Name Service Switch, which determines which name services (DNS, NIS, or local files) your host will use to look up information such as host names and Internet addresses. For more information, see the SAM online help.

*Restricted SAM for 10.10:*

- The superuser can specify a subset of SAM privileges and save it to a template file. The superuser can then grant non-superusers these privileges by loading a template file and saving it for the users. This saves time when granting numerous users this set of privileges.
- The superuser can specify a subset of SAM privileges for a group of users (as defined in the `/etc/group` file). When a user tries to run SAM, SAM first checks if the user has been granted privileges. If not, SAM will check if group privileges have been granted for the user's effective group id. This allows the superuser to quickly grant the same privileges to all users within a group.

---

## **SAM (System Administration Manager) for 10.0**

Features:

- Commercial security.

Additional and enhanced functionality is provided for password management and logon restriction functions. These changes meet and exceed the C2 level of security.

- 10.0 file system restructuring.

SAM fileset files reside in new locations that are determined by the 10.0 fileset structure.

- Logical Volume Manager.

There have been improvements to Series 800 Logical Volume configuration, and support has been added for Logical Volumes on the Series 700. Series 800 hard partitions are not supported by SAM on 10.x.

- SCSI Cascade support for the Series 700.

- Support for C1300/C2300 disk arrays.

- Device file naming and minor number changes.

SAM recognizes and creates device files as defined by the 10.0 I/O user interface definitions for device file names and minor number formats.

- Log file format and viewing changes.

Log file format and viewing changes allow flexible filtering and viewing of SAM log files.

- I/O convergence changes.

SAM complies with 10.0 changes in I/O commands for peripheral scanning, device file creation, and kernel building.

- Interactive addition of custom utilities to SAM.

You can add your favorite utilities or group of utilities to SAM interactively. These custom utilities can appear anywhere in SAM's list of areas.

- Non-superuser access to SAM functionality.

You can grant non-superusers limited access to SAM functionality or other custom utilities added to SAM. Where appropriate, these users are promoted to superuser while they are executing SAM functionality.

- Backward compatibility.

SAM in both 10.0 and 10.01 is not backward compatible to any 9.x based SAM add-ons (such as the X.25 functional area or the DCAM based CICS and Encina management tools). All user customizations to SAM will be lost when a system is upgraded from 9.x to 10.01. This includes additions to the Other Utilities menu, task customization for adding/removing users, and cluster clients, backup configurations, and exclusion lists used in the remove user action.

You should back up user customizations before you upgrade to 10.01. You can re-enter most of the customizations into SAM 10.01. However, in

## **5-66 Other Operating System and Subsystem Changes**



general, they will need to be modified to account for the 10.0 file system layout.

- SAM interface.

Some subareas of SAM have new or reorganized dialog windows as a result of new functionality. Also, the main menu used in SAM to launch individual functional area applications has been enhanced to display the functional areas of SAM as labeled icons. The textual form is still available as an option. For more information, refer to SAM's online help system.

For 10.01, the log file viewer has a search capability to search either forward or backward in the log file for a regular expression match.

---

## SCSI Pass-Through for Series 800 NIO Bus

SCSI pass-through (SCSI-PT) is a 10.01 development tool that enables a software developer to create code that communicates directly to a SCSI-2 device. SCSI-PT consists of a pass-through driver, an API to link the application software to the driver, and user reference documentation to explain the use of the product. With this product, an application can be created that directly passes SCSI commands to a device via the SCSI-PT API and driver and returns status to the application. To allow the direct control of the device, the following restrictions exist:

- Any device accessed via SCSI\_PT must be on a *standalone application bus*. Therefore, no boot, dump, or swap devices can be connected to this bus. While a SCSI\_PT application is running, it should have exclusive use of the application bus. There should be no other accesses to this bus, including from the operating system, a user, or another application.

The controlling application can access its device(s) via SCSI\_PT and other HP standard driver(s) as long as the application handles the coordination between the various drivers. Thus, the application is responsible for initiating all communications to device(s) on this bus. For example, a `tar` command cannot be directly used to a device on the exclusive application bus while that application is running. Any problems encountered must be displayed on this supported configuration (a standalone application bus) before support can be extended.

- If an HP device is being controlled by SCSI\_PT, the standard device diagnostics cannot be used. The SCSI\_PT driver does not support the standard device diagnostics, and if you want to use standard device diagnostics, you must use a kernel configured with the standard device drivers.
- HP SCSI-2 peripherals supported on HP platforms may not completely conform to the SCSI-2 standard.
- Not all HP SCSI devices have open command interfaces and may not support an open interface. Access to device command information may not be available for all HP devices.

Typical applications that use this driver are disk backup, system management, and tape archival applications.

---

## SharedPrint/UX

*For 10.10:*

This section describes SharedPrint/UX, a graphical user interface to the `lp` command. SharedPrint/UX provides these features:

- Easy-to-use interface with buttons for print options, such as double-sided and the number of copies.
- Ability to print most text and graphic files.
- Support for a wide range of hardcopy devices, such as laser printers, color inkjet printers, and large format plotters.

For more information, see the following information and the SharedPrint/UX online help (click **SharedPrint/UX** in the Help Manager). There is also more information, such as how to use SharedPrint/UX from the `lp` command line, in the *SharedPrint/UX User and Administrator's Guide for HP-UX 10.0* (B1171-90124).

## SharedPrint/UX in HP-UX 10.0 Versus HP MPower

The version of SharedPrint that comes with HP-UX 10.x does not support printing PostScript files on non-PostScript printers. This capability is available in HP MPower on HP-UX 9.x.

*For 10.20:*

HP recommends a stand-alone third-party PostScript rasterizer. The following is information on this product:

Product Name: ARC PRESS  
Contact Name: Environmental Systems Research Institute, Inc.  
380 New York St.  
Redlands, CA 92373-8100  
(800) 447-9778

---

<b>Note</b>	Support for this product is provided by Environmental Systems Research Institute, not Hewlett-Packard. Any problems with this product should therefore be reported directly to ESRI.
-------------	--

---

## Additional Printer Support

SharedPrint/UX supports these additional printers:

- DeskJet 1600C
- DeskJet 1600CM
- Laserjet 4V
- Laserjet 4VM
- Color Laserjet (PCL and Postscript)

## User Interface Changes

At 10.10, the Printers button in the SharedPrint/UX -Print dialog box has been modified. Clicking the button displays available printers in a scrollable list. Select the desired printer by double-clicking its name in the list, or single-click the name and click OK. The label of the Printers button will be updated with the printer you have selected.

## Setting Up SharedPrint/UX

SharedPrint/UX is not activated by default. You need to run the program `/opt/sharedprint/bin/sp_enable` to use SharedPrint/UX. Rerun this program whenever a new version of SharedPrint/UX is installed.

Restart the window manager even if the `sp_enable` program does not ask you to restart the VUE window manager.

To set up SharedPrint/UX, see the *SharedPrint/UX User and Administrator's Guide for HP-UX 10.0*. As the SharedPrint/UX manual indicates, part of the setup procedure involves choosing network support. You can either set up NCS, using the SharedPrint/UX manual, or you can set up HP DCE/9000 using these online release notes:

```
/opt/dce/newconfig/RelNotes/HPDCE1.3RelNotes.ps
/opt/dce/newconfig/RelNotes/HPDCE1.3RelNotes.txt
```

## Configuring a Jet Direct Printer

When configuring a Jet Direct printer for use with SharedPrint, the System Administration Manager (SAM) runs the `jetadmin` configuration program directly.

After the `jetadmin` main screen is displayed, follow these steps to configure SharedPrint for your printer.

1. Choose the option to configure a new printer.
2. Type the network address or network name of the printer.
3. Change the printer model script to `sharedprint`. For SharedPrint printers, the model script should always be set to `sharedprint`.

A printer name is assigned automatically. Record this name (the first item in the list) because you will need it later.

---

**Note** Because the automatically generated printer name might not be convenient to use, you can change the name. However, be sure that the name you choose is unique among all the printers on your network.

---

4. Choose the option to configure the printer.

## 5-70 Other Operating System and Subsystem Changes

5. When done, exit the `jetadmin` facility.
6. Exit SAM.
7. Run `/opt/sharedprint/sbin/setup_pr.sh`.
8. Type the printer name assigned in step 3 and click OK.
9. Choose the printer driver for your printer by clicking the labelled button next to the Printer Model/Interface button.

A list of supported printers is displayed. Choose the printer driver that corresponds to the type of printer you are configuring and click OK.

The Configure HP VUE Printers Subpanel dialog box is automatically displayed.

10. Add the printer to the Front Panel.
11. Click QUIT in the SharedPrint Setup dialog box.

You can now use SharedPrint to access this printer.

### **Using the SharedPrint/UX Dialog**

To display the SharedPrint/UX dialog, select a file in File Manager and choose Print from the Actions menu. You can also drop the file on the Printer control in the HP VUE Front Panel.

### **Printing with Default Options**

By dropping a file icon on an SP-labeled printer in the Printers subpanel, you can print with default options. These are the options selected when you last used the OK button on the Print window. When you print by using the Print button, then the Close button, no options are saved.

To see the print requests in the queue, click the Printer control in the HP VUE front panel.

In the Printers subpanel, the printer icons with an SP label are SharedPrint/UX printers. For information on adding more SharedPrint/UX printers to the Printers subpanel, refer to the *SharedPrint/UX User and Administrator's Guide for HP-UX 10.0*.

## Printing a Screen Image

1. Choose Capture Screen in the SharedPrint/UX dialog to print a window, a screen region, or the entire screen. For details, use the Help button on the Capture Screen window that appears.
2. From the Print window, print the screen image by choosing OK or Print.

---

## SharedX

*For 10.20:*

The SharedX receiver service (`/usr/sbin/recserv`) X authentication mechanism has been modified. This only affects the receiver service when it is run automatically by `inetd` in response to an attempt to share a window to a 10.20 system using SharedX. It does not affect the operation of the receiver service when it is run manually.

The receiver service has to put a dialog box on the receiver's display to ask if you want to accept a shared window. Prior to 10.20, it relied on "localhost" host-based authorization being enabled (for example, via the `xhost` command). This is the default for VUE.

With CDE, the default is to disable host-based authorization in favor of the `MIT-MAGIC-COOKIE-1` protocol. The receiver service has been modified to use an `XAUTHORITY` file created by the X server in `/var/spool/sockets/X11/user_name:display_number` (or `/tmp/.X11-unix/user_name:display_number`, if it exists). To give `recserv` access to this file, the `recserv` entry in `/etc/inetd.conf` has been modified so that `recserv` is run as root. Because `recserv` will be run as root, `/usr/lib/recserv` must also be owned by root to avoid a security hole. `recserv` will change its group ID and user ID to the group and owner of the `XAUTHORITY` file as soon as this file is located.

---

## Shells

This section covers the following topics:

- New Default sh
- POSIX Shell
- Bourne Shell
- Korn Shell

### New Default sh

At 10.0, the POSIX shell `sh-posix(1)` replaced the Bourne shell as the default shell. The POSIX shell now resides in `/usr/bin/sh`.

The POSIX shell is a superset of the Bourne shell, so your `sh` scripts in most cases will not need to be modified to take account of the change. But see “Bourne Shell” for exceptions and more information.

The Bourne shell is also delivered as part of 10.x, as `/usr/old/bin/sh`, so you can continue to use it if you prefer.

`sh-posix` will *not* replace the Korn shell, but it was the standard HP-UX shell as of 10.0, and Korn shell users as well as Bourne shell users are encouraged to migrate to it.

See “Differences between the Bourne and POSIX Shells” and “Differences between the Korn and POSIX Shells”.

### POSIX Shell

#### Logging

If you are root, your login shell is `/sbin/sh` or `/usr/bin/sh`, and `HISTFILE` is not set in `/.profile`, logging is turned off. That is, no history file is created.

With logging turned off, non-interactive application performance is improved. This is especially true for applications with multiple processes that access the same history file.

However, be aware that when logging is turned off, the following occurs:

- History-dependent functions, such as the `history` command, and command editing using `vi` and `emacs` will not work.

- Active job names will be unknown on the `jobs` command.
- Some scripts might not work as expected.

If you need the history functionality and compatibility, set the `HISTFILE` variable in `/.profile`.

*For 10.0:*

The OSF POSIX shell was ported into HP-UX 10.0 as `sh-posix(1)`.

- Replaces Bourne shell:
  - Delivered as `/usr/bin/sh`.
  - Directory `/bin/posix` and file `/bin/posix/sh` eliminated.
- Supports 4-byte EUC.
- Now conforms to XPG4 specifications.
- Three new or changed commands: `hash`, `type`, `export`.
- Position of reserved file descriptors changed.

New Commands and Options:

- “hash” becomes a shell built-in command with the addition of `-r`.
- New built-in command `type`.
- `export -p` will be fully implemented.

Reserved File Descriptors:

The reserved file descriptors 10 and 14-20 have been moved to the maximum side of the array of available file descriptors. This should not affect current users.

## **Bourne Shell**

The Bourne shell was replaced in 10.0 by the

POSIX shell, known from 10.0 on as `sh(1)`. The Bourne shell will not be enhanced in the future.

HP strongly encourages Bourne shell users to switch to the POSIX shell. But for those who cannot do without it, the Bourne shell is available on 10.x as `/usr/old/bin/sh`; these users can change their `PATH` variable to include `/usr/old/bin` before their regular path.

## **5-74 Other Operating System and Subsystem Changes**



## Differences between the Bourne and POSIX Shells

Although the POSIX shell is a superset of the Bourne shell and contains all the Bourne shell's syntactic constructs, Bourne shell users will notice a few changes, including the following.

- Positional parameters.

When a function call returns, the POSIX shell restores the positional parameter list. This does not happen in the Bourne shell, where the positional parameters are global to the functions.

For example, if a `shift` operation is performed within a function invoked by the POSIX shell, its effect will not be felt after the function returns.

- Errors in functions.

These abort the function but not the script.

- Lookup order.

Function lookup is done before the built-in commands.

- Error messages.

Some error messages output by the POSIX shell are different from Bourne shell messages.

- `LC_TYPE`, `LANG`

Changing the value of `LC_CTYPE` or `LANG` after the shell has started will not affect the lexical processing of shell commands in the current shell execution environment or its subshells.

- `IFS`

The shell treats each character of `IFS` as a delimiter.

1. If the value of `IFS` is

`<space><tab><newline>`

or if it is unset, then:

- Any sequence of those characters at the beginning or end of the input is ignored.
- Any sequence of those characters within the input delimits a field.

(For example,

```
<newline><space><tab>foo<tab><tab>bar<space>
```

results in two fields, `foo` and `bar`.)

2. If the value of `IFS` is null, no field-splitting occurs.
3. Otherwise the following rules are applied in sequence, the term *IFS white space* meaning any sequence of white-space characters that are in the `IFS` value (for example if `IFS` contains

```
<space><comma><tab>
```

then any sequence of space and tab characters constitutes *IFS white space*).

- a. *IFS white space* at the beginning and end of input is ignored.
- b. Each occurrence in the input of an `IFS` character that is not *IFS white space*, along with any adjacent *IFS white space*, delimits a field.
- c. Non-zero-length *IFS white space* delimits a field.

■ **select**

This command does not exist in the Bourne shell.

**POSIX Shell Features New to Bourne Shell Users**

New capabilities that users switching from the Bourne shell will find include:

■ **New built-in commands.**

Examples: `whence`, `alias`, `fg`, `bg`, `jobs`, `fc`, `getopts`, `kill`, `let`, `unalias`.

■ **read, print**

These have a `-p` option for co-processes.

■ **New operators.**

Examples:

```
[[.]], ((...)), >|, and |&
```

■ **!**

**5-76 Other Operating System and Subsystem Changes**

Any command preceded by a “!” returns the logical negation of the exit status of the command.

- Reserved words.

Examples:

```
!  
selectd  
function  
[[  
]]
```

- Aliases.

Aliases can be used to re-define special built-in commands. Aliases can be created, listed and exported.

- New parameter substitutions.

In addition to the substitutions allowed in the Bourne shell, substitutions can be performed with the following:

```
${name#pattern}  
${name##pattern}  
${name%pattern}  
${name%%pattern}
```

- One-dimensional array facility.

An element of an array is referenced by a subscript. `set -A` can be used to assign values to an array.

- Tilde substitution.

Tilde substitution can be performed.

- Parameters.

In addition to parameters such as `#`, `-`, `?`, `$` that are set automatically in the Bourne shell, the following parameters are set by the POSIX shell: `ERRNO`, `LINENO`, `OLDPWD`, `OPTIND`, `OPTERR`, `OPTARG`, `PWD`, `PPID`, `REPLY`, `RANDOM SECONDS`.

Similarly, in addition to the parameters such as `IFS` and `CDPATH` that are used by the Bourne shell, the following parameters are used by the POSIX shell: `EDITOR`, `ENV`, `FCEDIT`, `VISUAL`, `PS3`, `PS4`, `HISTFILE`, `HISTSIZE`.

- Positional parameters.

Positional parameters are not limited to 9 as in the Bourne shell.

- Integer arithmetic.

Integer arithmetic can be performed using the `let` command. Or you can use the following construct:

```
((...))
```

- New input/output operators.

These include:

```
>|, <>, <&p, and >&p
```

- Single-step assignment.

Values can be assigned in a single step using `readonly` and `export` (for example `export LANG=french`) instead of the two steps needed in the Bourne shell.

- Symbolic names.

Symbolic names for signals and traps can be used.

- Environment variables automatically inherited.

An environment variable is passed to the child process even if it is not exported.

- `for`, `while`, `until`

`for`, `while`, and `until` loops are executed in the current process environment. Assignments made within loops remain effective even after the loop completes.

- Job control.

Job control capabilities allow the user to suspend, resume, and control processes executing in the background.

- Command-line editing.

## 5-78 Other Operating System and Subsystem Changes

You can do command-line editing with `vi` or `emacs` editing commands.

- History.

You can use a history file to list and edit command lines.

## Korn Shell

The Korn shell continues to be supported on 10.x as `ksh(1)`, although the POSIX shell is now the standard HP-UX shell, and HP encourages Korn shell users to switch to it.

*Changes at 10.0:*

The reserved file descriptors 10 and 14 through 20 have been moved to the maximum side of the array of available file descriptors. This is a bug fix and should have no impact on current users.

## Differences between the Korn and POSIX Shells

Although the POSIX shell is based on the Korn shell, there are several differences between them.

The following is a summary of POSIX shell features that do not exist in the Korn shell, or are implemented differently.

- `! command(s)`

Negates the exit status of the `command(s)` just executed.

- `command [arg ...]`

Treats `arg` as a command, but disables function lookup on `arg`.

- `export -p`

Writes to the standard output the names and values of all exported variables, using quotes so that they can be re-input to the shell as commands that export the same variables in the same way.

- `getopts optstring name [ arg ... ]`

`name` contains `?` when end of options is found.

- `hash [external commands/programs]` or `hash -r`

The `hash` command adds external command locations to its list of remembered locations. When no arguments are specified, it reports on the contents of the list.

The `-r` option removes the locations of all previously remembered external commands.

■ `kill -l`

Lists signal names.

■ `kill -L`

Lists signal numbers and names. Produces the same output as `kill -l` in the Korn shell.

■ `readonly -p`

Writes to the standard output the names and values of all read-only variables, using quotes so that they can be re-input to the shell to set the same attributes to the same values.

■ `set +`

Lists the names of all shell variables without their values.

■ `set -C`

“Noclobber.” Can also be set (with `-C`) when the shell is invoked.

■ `test -e file` or

```
[[ -e <file> ]]
```

True if file exists.

■ `type`

Indicates how each argument would be interpreted if used as a command name.

■ `umask -S`

Prints the current value of the mask in symbolic format.

■ `unalias -a`

## 5-80 Other Operating System and Subsystem Changes

Removes all the alias definitions from the current shell execution environment.

■ **unset -v *names***

“Unassigns” *names* (erases their values and attributes). If the **-v** option is used, *names* refer to variable names.

Unsetting **ERRNO**, **LINENO**, **MAILCHECK**, **OPTARG**, **OPTIND**, **RANDOM**, **SECONDS**, **TMOU**T and **\_** removes their special meaning even if they are subsequently assigned to.

■ **whence -p *name***

Output is different from **ksh**.

The **-p** option does a path search for *name* even if *name* is an alias, a function, or a reserved word.

■ **IFS**

See “Differences between the Bourne and POSIX Shells” earlier in this chapter for a description of **IFS** in **sh-posix**.

■ **OPTERR**

New **sh-posix** parameter.

■ **Lookup order:**

- *sh-posix*: alias function built-in \$PATH
- *ksh*: alias built-in function \$PATH

■ **-c**

If **sh-posix** is invoked with **-c**, a process will be spawned for it:

```
$ /usr/bin/sh -c "ps -f"
UID  PID  PPID  C STIME  TTY  TIME COMMAND
usera 511  510   3 16:15:46 tttyp8  0:00 ps -f
usera 501  500   0 16:15:36 tttyp8  0:00 -ksh
usera 510  501   1 16:15:45 tttyp8  0:00 /usr/bin/sh -c ps -f
```

■ **Signals**

**sh-posix** has the following signals:

- 0) NULL
- 6) ABRT (IOT for ksh)

■ Tilde expansion

`sh-posix` provides tilde expansion on the fly for the following parameter substitutions:

```
    ${parameter:-word}
    ${parameter:+word}
    ${parameter:=word}
    ${parameter:?word}
```

For example, if `~` is `/users/usera`:

```
$ a=${strings:-~}

a --> /users/usera (sh-posix)
a --> ~             (ksh)
```

■ Exit values for failed `exec`.

`sh-posix` exits with:

- 126 for failed `exec` with `ENOEXEC`
- 127 for failed `exec` with `EACCESS`

■ Positional parameter 0

When executing a function, `sh-posix` leaves positional parameter 0 unchanged. `ksh` sets this parameter to the function name.

Features in `ksh` that are not in `sh-posix`:

■ Keyword `time`

Does not exist in `sh-posix`.

■ `_`

Is not a special variable in `sh-posix`, so no special processing is done.

## 5-82 Other Operating System and Subsystem Changes



---

## Small Disks and Memory

### Small Disk Configuration

HP-UX 10.x is structured such that it allows for smaller disks. The kernel filesets have been split so that only those things necessary for building the kernel are in them.

- The filesets containing libraries and utilities needed to build a kernel have the suffix **-KRN**.
- The filesets containing software needed to develop applications have the suffix **-PRG**.
- The filesets containing software needed to run the system (not to build the kernel or to develop any programs) have the suffix **-RUN**.

As a result, you have the option of removing the **-KRN** and **-PRG** filesets once you have a kernel you like. However, if you were to remove those filesets and later wanted to make any kernel modifications, you would need to reinstall the **-KRN** filesets.

### Small Memory Configuration

If you order the optional Desktop HP-UX product, a new “light-weight” kernel/system configuration called the Lite HP-UX Configuration is included by default. The Lite HP-UX Configuration is specifically tuned for single-user commercial HP-UX user using a desktop workstation with limited disk space and/or RAM. The kernel is smaller than the standard 10.x kernel because it has fewer drivers and some kernel tunable parameters set to different values. In addition, the amount of memory consumed by user processes is reduced because several non-critical subsystems/daemons are not started by default.

---

<b>Note</b>	The Lite HP-UX Configuration is <i>not</i> intended to support servers, multi-user systems, centralized network administration tasks, and high-end technical workstations.
-------------	--

---

Consult your HP Sales Representative (SR) for details.

## FreeDisk

The `freedisk(1M)` tool can be used to identify and remove unused filesets if other software does not depend on them. `freedisk` has been included in previous Series 700 releases, but starting with HP-UX 10.0, it is provided on both Series 700 and Series 800 computers.

The pre-10.0 previous version of `freedisk` was modified at HP-UX 10.0 as follows:

- It now resides in `/usr/sbin/freedisk`.
- You can use `freedisk` to remove the kernel build libraries. If your kernel will not be rebuilt for any reason, you can recover significant amounts of disk space by removing the filesets related only to building a kernel.
- `freedisk` functionality that was included in the System Administration Manager (SAM) is not included in 10.x. Specifically, `freedisk` can be used on 9.0 to find and remove log files, core files, and other large, unused files. However, to do this on HP-UX 10.x, you should instead go to “Routine Tasks” in the top level SAM menu.

For more information on `freedisk`, refer to the `freedisk(1M)` manpage.

## Network Font Services

You can save disk space on your 10.x workstation by configuring HP VUE to access fonts from a network font server (refer to the `mk_fnt_clnt(1M)` manpage). And you can configure your system as a font client the first time you start up your workstation. A font client runs the VUE 3.0 environment without having to have the X11 font filesets on disk, which saves about 8 MB of disk space. The font client accesses the X11 fonts over the network from a font server.

The following are some restrictions to setting up a font client:

- Your system must not contain the SharedPrint Server filesets. By default, Runtime (but not Desktop) contains the SharedPrint Server filesets.
- Diskless servers and diskless clients are not supported as font clients. Before configuring an existing font client to be a diskless server, you should restore the X11 font filesets and cease being a font client.

## 5-84 Other Operating System and Subsystem Changes

- In general, any server system providing X11 or HP VUE resources to other clients (such as a system serving X terminals or an MPower server) should not be a font client.
- If your system uses something other than HP VUE 3.0 for its X11 interface, you should not make the system a font client.

---

## System Calls

### **poll()**

The poll system call, `poll()`, is provided on both Series 700 and Series 800 computers for 10.x. On pre-10.0 HP-UX releases, this system call is available on Series 700 computers and on Series 800 computers that have the Streams subsystem configured. `poll()` provides a general mechanism for reporting I/O conditions associated with a set of file descriptors and waiting until one or more specified conditions becomes true. Specified conditions include the ability to read or write data without blocking and error conditions. The programming interface is documented in the `poll(2)` manpage.

### **pstat()**

Additional fields that were previously only available on the Series 800 are available on the 10.x Series 700. Also, these fields no longer require access through `/dev/kmem`. For more information, refer to the `pstat(2)` manpage.

### **vfork()**

The 9.0 Series 700 internal code structure and performance characteristics are provided on 10.x Series 800 computers. Note that `vfork()` exists on 9.0 Series 800 computers, but it is implemented as `fork(2)`. The programming interface has not been changed. For more information, refer to the `vfork(2)` manpage.

---

## typedef Changes

*For 10.20:*

Several system typedefs are changed as part of the development for future releases. In most cases, the size of the type did not change. If the size did change, the change is an increase in size. All customer-supported typedefs remain the same size.

---

## UUCP

*For 10.10:*

---

**Note** Release 10.10 will be the last release that UUCP configuration is supported in SAM. In the next release, UUCP will continue to be supported on HP-UX, but SAM will not help configure it.

---

*For 10.0:*

- Complies with 10.0 file system layout.
- Complies with NFS Diskless File Sharing Paradigm.
- Supported over TCP/IP; new file `/usr/sbin/uucpd`.
- Conversion scripts no longer supported.

Each of these changes is discussed in the subsections that follow.

### Complies with 10.0 File System Layout

UUCP files have been restructured as follows:

Type			Private
of File	10.0 Layout	9.x Layout	/Shared
=====			
UUCP			
config.	<code>/etc/uucp</code>	<code>/usr/lib/uucp</code>	P
Startup/			

shutdown config.	/etc/rc.config.d	(did not exist)	P
Log files	/var/adm	/etc	P
UUCP log, error files	/var/uucp	/usr/spool/uucp	P
UUCP spool dir.	/var/spool/uucp	/usr/spool/uucp	P
UUCP lock files	/var/spool/locks	(did not exist)	P
UUCP data transfer (default)	/var/spool/ uucppublic	/usr/spool/ uucppublic	P
Execut- ables	/usr/bin	/usr/bin	S

Type of File	10.0 layout	9.x layout	Private /Shared
=====			
System admin. commands	/usr/sbin	/etc	S
UUCP backend execut- ables	/usr/sbin/uucp	/usr/lib/uucp	S
Startup/ shutdown scripts	/sbin/init.d	(did not exist)	S

For more information on private and shared directories, see “The HP-UX 10.0 File System Layout” in Chapter 4.

### **/usr/include/dial.h Changes**

The include file `dial.h` contains the definition of the UUCP configuration files. As of 10.0, this file was changed as follows:

- The variables `SYSTEMS`, `DEVICES`, `DIALERS`, and `DIALCODES` have been redefined to identify the new locations of UUCP configuration files.

These are `/etc/uucp/Systems`, `/etc/uucp/Devices`, `/etc/uucp/Dialers`, and `/etc/uucp/Dialcodes` respectively.

An application that includes `dial.h` and uses any of these variables needs to be recompiled for 10.x.

- The variables `LDEVS`, `SYSFILE`, and `DIALFILE`, which corresponded to `/usr/lib/uucp/L-devices`, `/usr/lib/uucp/L.sys` and `/usr/lib/uucp/L-dialcodes` respectively, have been removed.

Although UUCP Version 2 was no longer supported in 9.x, the 9.x version of `dial.h` still referenced these Version 2 UUCP configuration files.

## **5-88 Other Operating System and Subsystem Changes**

Replace references to these variables with references to `DEVICES`, `SYSTEMS`, and `DIALCODES`, respectively (these point to the UUCP HoneyDanBer versions of the files at the locations indicated under the previous bullet).

### **Complies with NFS Diskless File Sharing Paradigm**

The NFS Diskless file sharing paradigm has the following implications for UUCP:

- UUCP context-dependent files (CDFs) removed
- All the commands, including `ct` and `cu` can be run from any system: standalone, client or server.
- Commands access the physical ports and associated device files on the system (standalone, client or server) on which the command is issued.
- Each system, standalone, client or server, has its private UUCP configuration, Log files, Lock, Spool and Working files.
- The only files shared between server and clients are the command executables. Shared and private files are accessed from all systems via NFS.

To run UUCP on an NFS Diskless server from an NFS Diskless client, telnet to the server and then run UUCP.

### **Supported over TCP/IP**

As of 10.0, UUCP was supported over TCP/IP; a UUCP `inet` daemon (`/usr/sbin/uucpd`), answers requests on a user-definable TCP port.

The `inet` configuration consists of two files, `/etc/services` and `/etc/inetd.conf`.

You need to add the following entries:

- `/etc/services`:

```
uucp          540/tcp  uucpd          # uucp daemon
```

(where 540 is the user TCP port).

- `/etc/inetd.conf`:

```
uucp    stream tcp nowait root /usr/sbin/uucpd uucpd
```

You can also add an optional entry in `/var/adm/inetd.sec` to enable security restrictions.

When you have made these changes, restart the `inetd` daemon and force it to re-read `/etc/inetd.conf` by issuing the command

```
/usr/sbin/inetd -c
```

On the UUCP calling side, UUCP configuration must be set up for use with TCP/IP. For example:

```
/etc/uucp/Devices: TCP - - - TCP
```

```
/etc/Systems: <system> Any TCP - - ogin: uucp ssword: <passwd>
```

### **Conversion Scripts No Longer Supported**

The scripts to convert Version 2 UUCP files (`L-devices`, `L-dialcodes`, `L-cmds`, `USERFILE`, `L.sys`) to HoneyDanBer UUCP files are no longer supported. (HoneyDanBer is also known as Basic Networking Utilities, BNU).



---

## VUE (HP VUE 3.0)

Unless indicated otherwise, all subsections describing VUE are applicable to 10.20.

### HP VUE 3.0 and MPower Announcement

*For 10.20:*

HP-UX 10.20 is the last release of HP VUE 3.0 and MPower 2.03; these products will not be available in future HP-UX releases.

### HP VUE 3.0 and Large User IDs

*For 10.20:*

The HP VUE product on 10.20 does not support large UIDs. The CDE desktop product should be used on systems where large UIDs will be utilized.

### HP VUE 3.0 Troubleshooting and Miscellaneous Changes

#### Diagnosing VUE Session Startup Problems

If your VUE session fails to start, the problem could be that something failed during the execution of the Xsession script. Significant Xsession events for the current login session are noted as they happen in `$HOME/.vue/startlog`. If you examine this file, you may be able to determine the point of failure.

Also, beginning with 10.01, run levels are generally higher by one to perform the same function; for example, initiating run level 4 starts VUE. For more information, see Chapter 1 of the *HP-UX System Administration Tasks* manual.

#### Hostname Aliases and HP VUE

The HP Visual User Environment (VUE) will not work properly if the system's hostname is set to an alias. If the hostname is set to an alias, VUE will not be able to start the messaging system it needs and you will not be able to use VUE. To prevent this problem, do not set the system's hostname to an alias if VUE is to be used on the system.

## Upgrading Users from VUE 2.01 to VUE 3.0

*For 10.20 and 10.10:*

HP VUE 3.0 was introduced with the HP-UX 9.x release. This environment contained tools for migrating users from their personal VUE 2.01 environment to VUE 3.0; this procedure is documented in the chapter of the *HP Visual User Environment 3.0 User's Guide* (part number B1171-90061) titled "For Previous HP VUE 2.01 or X11 Users".

HP-UX 10.20 and 10.10 continues to offer HP VUE version 3.0. However, VUE on HP-UX 10.x no longer contains the upgrade tools that help users migrate from VUE 2.01 to VUE 3.0. All VUE users must convert their environment to VUE 3.0 using the migration tools provided on HP-UX 9.x before upgrading to an HP-UX 10.x system.

VUE 3.0 on 10.x no longer silently accommodates VUE 2.01 user configurations and serious loss of user customizations can occur if a VUE 2.01 user environment is carried forward into the HP-UX 10.x VUE 3.0 environment.

### Getting VUE Running on Non-Default Displays

If you have a PCXT system (Series 715/33, 50, 75 or 725/50, 75) and you have an external graphics device (such as the VISUALIZE-24 or the CRX48Z) in addition to the internal device that comes with the system, you need to make some configuration changes to get the X server and VUE to come up using the external graphics device as the default screen.

To do this, you must edit the `/etc/X11/X0screens` files. This file tells the X server which "screen/display" to come up on. In the PCXT systems, this file points the server to the internal graphics device. You need to change this to reference the external graphics device. To do this, change the line:

```
Screen /dev/crt
```

to:

```
Screen /dev/<external device location>
```

where `<external device location>` is the device file that corresponds to the external graphics device.

If you have a non-PCXT system (Series 715/64, 80, 100, or 725/100), you do not need to make this change. These systems automatically configure the

## 5-92 Other Operating System and Subsystem Changes

additional graphics device to be your default screen, and the X server and VUE will come up using that external graphics device.

### **X11R4/Motif 1.0/1.1 Application Font Usage**

HP VUE is now built on X11R5 and Motif 1.2. Motif 1.2 extended the concept of font lists to allow a font list to contain either a font struct, a font set (a new X11R5 structure used for internationalized applications), or any mixture of font structs and font sets.

Use of a font set by an application makes sure that the font selected for that application matches the encoding of the data generated by that application. For example, if the application is run in a locale that uses the ISO 8859.1 character set, it is important that an ISO 8859.1 font be used to render that data. Further, some languages require more than one font to correctly render all characters. The concept of font sets is used to open multiple fonts for these languages, treating them as a single entity.

Since HP VUE is made up of internationalized clients, vuestyle has been changed to generate font list resources that utilize the internationalization capabilities provided by Motif 1.2 and X11R5. This means that Motif 1.2 clients will get internationalized font resources generated by vuestyle.

Applications built archived with Motif 1.1 or Motif 1.0 will experience problems with the new font list resources generated by vuestyle. These applications may obtain the default fixed font, or in extreme cases, may fail completely. Several simple workarounds exist for these clients:

- Font settings in the resource environment can be targeted specifically to those particular applications that have problems with the vuestyle exported font set settings. This can be done by creating or appending to an application's app-defaults file with the old-style font list resource specifications.

This allows new Motif 1.2/X11R5 clients to still obtain the benefits of internationalized font resources under HP VUE while allowing the older clients to operate as they have in the past.

Changes to font resources made by vuestyle will not affect clients that use this workaround.

- Modify vuestyle's app-defaults file. In this file, the old 1.1 vuestyle font list resources are provided, but are commented out. Simply uncomment the old resources, and comment out the new resources. Then, bring up vuestyle and select a font; save the session as your home session or restore it as your current session to cause the change to be reflected for future sessions.

Using this workaround has the benefit that your old clients will still respond to font changes made with vuestyle. The disadvantage is that new X11R5/Motif 1.2 clients will not obtain the internationalization functionality from the fonts that they expect. Those clients will still work, but they may obtain the wrong font (that is, use a Roman8 font when an ISO 8859.1 font is required).

## **Vuelogin**

*For 10.10:*

The vuelogin component has been modified to allow a user to log in when their home directory is unavailable. For example, this might occur when the user's home directory is NFS mounted and the NFS server is unavailable.

If a user's home directory is inaccessible at login, the user is provided with a failsafe session. The current working directory of the user is changed from `$HOME` to `/`.

Note that the `$HOME` variable is assigned the user's home directory even if the directory is inaccessible.

## **Vuelogin 'langSetup' Resource**

*For 10.10:*

A new resource, langSetup, has been added to vuelogin. This resource points to a script that will be executed whenever the user selects a different locale from the login screen's language options. The default setting for langSetup is the script `/etc/vue/config/Xlangsetup`.

This script checks the newly selected language and, if required, will install the appropriate keymap to the X server.

For more information, refer to the `vuelogin(1)` manpage.

## **5-94 Other Operating System and Subsystem Changes**

## **Support for New Locales in VUE**

*For 10.10:*

Support was added for the following locales as part of the fileset VUE-RUN:

- ar\_SA.iso88596 - Arabic
- bg\_BG.iso88595 - Bulgarian
- cs\_CZ.iso88592 - Czech
- el\_GR.iso88597 - Greek
- hr\_HR.iso88592 - Croatian
- hu\_HU.iso88592 - Hungarian
- iw\_IL.iso88598 - Hebrew
- pl\_PL.iso88592 - Polish
- ro\_RO.iso88592 - Rumanian
- ru\_RU.iso88595 - Russian
- sk\_SK.iso88592 - Slovakian
- sl\_SI.iso88592 - Slovenian
- tr\_TR.iso88599 - Turkish

## **X11 New Bitmap Fonts**

*For 10.10:*

New fonts were added for the following codesets:

- iso8859-2
- iso8859-5
- iso8859-7
- iso8859-8
- iso8859-9

Each of the codesets consists of eight user (fixed space) fonts and seven system (proportional) fonts.

## **X11 FontServer for 10.20**

*For 10.20:*

**TrueType Fonts.** The same set of fonts available in Intellifont format are now available in TrueType format. Hewlett-Packard will discontinue support for AGFA's Intellifont font technology in a future release.

The Intellifont fonts are provided in an updated format that can be handled with the UFST rasterizer. Some Intellifont fonts might not be provided in future releases. However, the same font name would be supported using a `fonts.alias` file.

**Administering Charsets for True Type Fonts.** Character set definitions for True Type fonts are stored in the directory `/usr/lib/X11/fonts/stadmin/ttf/charsets`. Files in this directory named `characterset.sym` define the character mapping for a particular character set.

**Wildcard Aliases.** HP provides an extension to support wildcard aliases. If a `fonts.alias` file contains the line

```
WILDCARD_ALIASES_SUPPORTED
```

an alias name can contain wildcard entries. For example, the entry

```
-agfa-univers-medium-r-normal--10-*-*-p*-hp-roman8 \  
-adobe-utopia-medium-r-normal--10-100-75-75-p-60-hp-roman8
```

will cause all of the following font descriptions to reference the adobe-utopia font instead of the agfa-univers font:

```
agfa-univers-medium-r-normal--10-*-*-p*-hp-roman8  
agfa-univers-medium-r-normal--10-72-100-100-p-0-hp-roman8  
agfa-univers-medium-r-normal--*-72-100-100-p*-hp-roman8
```

## **X11 FontServer for 10.10**

*For 10.10:*

The HP-UX 10.10 fontserver is based on X11 R6. New fontserver capabilities include:

- R6 mechanisms for font manipulation
- R6 character encoding set
- Glyph caching and scalable aliases
- Universal Font Scaling Technology (UFST) rasterizer

## **5-96 Other Operating System and Subsystem Changes**

- TrueType fonts

This section provides a short description of the new R6 fontserver capabilities. For complete information, refer to the *Programmer's Supplement for R6* (ISBN 1-56592-089-9) by O'Reilly & Associates, Inc.

**Font Manipulation.** New R6 mechanisms are used to specify anamorphic scaling, obliquing, mirroring, and rotation of fonts.

The new mechanism uses a set of four numbers delineated by brackets to replace the `pointsize` or `pixelsize` fields. It is used to create a two-dimensional matrix that controls the transformation for each character.

Future releases of the fontserver will not support the following Hewlett-Packard proprietary XLFd extensions:

- `slant` (obliquing)
- `addstylename` (mirroring and rotation)
- `pixelsize` and `pointsize` (anamorphic scaling)

The Hewlett-Packard proprietary `weightname` extension used to specify darker or lighter fonts will be supported.

**Charset Encoding Extension.** HP-UX 10.10 supports the X11 R6 charset encoding extension. The charset encoding extension allows subsetting of fonts so that not all characters need to be generated for a font.

The Hewlett-Packard syntax for the charset encoding extension is obsolete and will be removed in a future version of the font server.

**Glyph Caching and Scalable Aliases.** Glyph caching and scalable aliases are standard capabilities in the X11 R6 fontserver. Glyph caching is the deferred loading of character glyphs and allows X to reduce memory and computation requirements associated with the generation of fonts.

Scalable aliases increase the capability of font name aliases to allow them to be used with scalable font names and with the matrix XLFd enhancement.

**New Rasterizer.** The new Universal Font Scaling Technology (UFST) rasterizer from AGFA replaces the `type1` and `intellifont` rasterizers. The UFST rasterizer supports `truetype`, `type1` and `intellifont` font files.

**New Fonts.** The 10.10 release includes a set of 35 truetype fonts. These include treatments from the following families of AGFA fonts:

- Albertus
- AntiqueOlive
- CGomega
- CGtimes
- ClarendonCondensed
- Coronet
- Courier
- Garamond
- LetterGothic
- Marigold
- Univers
- UniversCondensed

The Intellifont fonts are provided in an updated format that can be handled with the UFST rasterizer. Some Intellifont fonts may not be provided in future releases. However, the same font name would be supported using a `fonts.alias` file.

**Command Changes.** In a future release, the `-tfm` option will be removed from the `stmkdirs` command. These metrics are no longer used and are not generated for truetype fonts.

#### **Image Help When Using Localized Environments with HP VUE**

If you install Imaging help for a localized environment, the help may not appear in the main help window. To fix this problem, log into a localized VUE session as `root` and run `helpgen` in a terminal window.

#### **Text Help When Using Roman8 Locales with HP VUE**

Translated text in the main help window for Roman8 locales (locale names ending with `.roman8`) that contain accents are not displayed correctly. To fix this problem, change the links to the help browser volume for the help browser. As `root`, execute the following:

```
cd /etc/vhelp/volumes/[locale]  
rm browser.hv
```

### **5-98 Other Operating System and Subsystem Changes**



```
ln -s ../../../../etc/vhelp/help/[newlocale]/Browser/browser.hv .
```

where `[locale]` is the Roman8 locale name, and `[newlocale]` is the locale name with `.iso88591` in place of `.roman8`.

## HP VUE and Commercial Security (Trusted Systems)

*For 10.10:*

Commercial Security can be enabled on a system to take advantage of trusted system security features. In a trusted system, the encrypted password registry for users is located in `/tcb/files/auth/` rather than in generic password registry files such as `/etc/passwd` and `/etc/group`.

In addition, the protected password database files in a trusted system contain various fields that can be used to restrict access for any user account.

HP VUE uses several components to authenticate users when they log into the system. These components are `vuelogin`, `vuegreet`, `vuession` and `vuelock`. On a trusted system these VUE components do use the protected password databases to authenticate users. However, the VUE components do not support monitoring of user accounts based on the database account restrictions. Security is therefore not fully implemented on such a system.

If you intend to convert a system that runs HP VUE automatically (the default VUE behavior), you may want to reconfigure your system to disable the automatic startup. To do so, edit the `/etc/inittab` file and change the run level from 4 to 3 in the entry: `init:4:initdefault:.`

You can also remove the fileset `VUE-RUN` to make sure that none of HP VUE components are manually started. Refer to the `swremove(1M)` manpage for information about how to remove a fileset on your system.

## Changes in Fileset Organization

*For 10.20 and 10.10:*

The X Window System consists of three products: X11, VUE, and CDE. The following changes have been made to X Window System filesets:

- For 10.20, the `X11-RUN-MAN` fileset has been subdivided into `X11-RUN-MAN`, `X11-RUN-CT-MAN`, and `X11-RUN-CL-MAN`, which corresponds to the filesets `X11-RUN`, `X11-RUN-CTRB`, and `X11-RUN-CL`.

- The 9.x filesets `X11-FONTA`, `X11-FONTB`, and `X11-FONTC` have been combined into 10.x filesets `X11-FONT-MIN` and `X11-FONT-AUX` in the X11 product.
- The 9.x fileset `X11-RUN` has been subdivided into 10.x filesets for the X11 and VUE products. For 10.20, the X11 product contains the filesets `X11-RUN`, `X11-RUN-CTRB`, and `X11-RUN-CL`. For 10.10, the X11 product contains the filesets `X11-RUN`, `X11-RUN-AUX`, and `X11-RUN-CL`. `VUE-RUN` and `VUE-RUN-AUX` are part of the VUE product.
- The 9.x fileset `X11-RUN-HELP` has been renamed to 10.x fileset `VUE-HELP-INFO` in the VUE product.
- Filesets `X11-RUN-MAN`, `X11R5-PRG-MAN`, `X11R5-SHLIBS`, and `X11R5-PRG-MAN` have each been partially split up, creating the new filesets `VUE-MAN`, `VUE-PRG-MAN`, `MOTIF12-SHLIB`, `MOTIF12-PRG`, and `VUE-HELP-PRG`. `VUE-PRG-MAN`, `MOTIF12-PRG`, and `VUE-HELP-PRG` are delivered in the HP-UX Developer's Toolkit.

---

**Note**           HPterm has been moved from the `VUE-RUN` fileset to the `X11-RUN-CL` fileset.

---

### **Toggling Word Wrap Mode in Text Editor**

*For 10.20:*

This section is not applicable for 10.20.

*For 10.10:*

In Text Editor (`vuepad`), toggling Word Wrap mode before typing a line of text that includes a new line will cause Text Editor to abort unexpectedly. This will be fixed in a future Motif library patch.

### **Use of `/etc/profile` During VUE Session Startup**

Several standard environment variables (including but not limited to `PATH`, `SHLIB_PATH`, and `MANPATH`) are typically defined in `/etc/profile` for a `getty` login. VUE now leverages this same mechanism by, if possible, sourcing `/etc/profile` from within the Xsession script. `/etc/profile` is considered sourceable if `grep $VUE /etc/profile` evaluates true, which means that since

## **5-100 Other Operating System and Subsystem Changes**

the script contains a reference to `$VUE`, it can safely be run in VUE's GUI environment with `VUE=true`.

All `/etc/profile` `PATH`, `SHLIB_PATH`, and `MANPATH` definitions are first appended to those set up by `vuelogin`, duplicate path elements are weeded out, and finally invalid path elements are removed. The `/etc/profile` processing happens in `Xsession` ahead of `<auto-customize>`, `.vueprofile`, and `Xsession.d` processing.

### Functions Marked for Obsolescence

*For 10.20:*

This section is not applicable for 10.20.

*For 10.0:*

XHP functions for internationalized keyboard input will be removed at the next release of HP-UX. These HP proprietary functions are no longer needed since X11R5 provides public, standard APIs that provide equivalent functionality. Specifically, the following functions will be removed with the next release of HP-UX:

```
XHPConvertLookup()
XHPGetEurasianCvt()
XHPInputChinese_s()
XHPInputChinese_t()
XHPInputJapanese()
XHPInputKorean()
XHPInputRoman8()
XHPInputIS07sub()
XHPNlioct1()
```

Instead, applications should use the following X11R5 routines:

```
XOpenIM()
XCloseIM()
XGetIMValues()
XCreateIC()
XDestroyIC()
XSetICFocus()
XUnsetICFocus()
```

```
XmbResetIC()
XIMOfIC()
XSetICValues()
XGetICValues()
XmbLookupString()
XwcLookupString()
```

The new X11R5 functionality gives applications greater control and more flexibility than the HP proprietary functions.

Additionally, the function `XHPis16BitCharacter` has been removed. The function is meaningless in an environment where characters can be 24 or 32 bits in size.

### **Changes in Keyboard Functionality**

*For 10.20:*

This section does not apply for 10.20.

*For 10.0:*

The `XHPSetKeyboardMapping` function and the related routines `XHPRefreshKeyboardMapping()` and `XHPSetKbdMapInit()` will be removed at the next release of HP-UX. These functions are used to allow an application to change their local copy of the keymap to emulate a particular national language keyboard. The functions only work for HP HIL keyboards. Future HP workstations may no longer support HP HIL keyboards, so these functions have a diminishing value. An application or user cannot depend on the functions working in all cases.

In HP-UX 9.x, many applications (such as `vuepad` and `hpterm`) call `XHPSetKeyboardMapping`, passing through the value of `KBD_LANG` environment variable to indicate the keyboard language to be used by the application. As of 10.0, HP VUE no longer set the `KBD_LANG` environment variable. Clients and libraries have removed the call to `XHPSetKeyboardMapping()` since its behavior is not reliable in a mixed DIN and HIL keyboard environment.

### **Rendering International Characters with 10.0**

*For 10.20:*

## **5-102 Other Operating System and Subsystem Changes**

This section does not apply for 10.20.

*For 10.0:*

In X11R3 and X11R4, support for rendering Asian characters was done through a proprietary mechanism called the associate font mechanism. Through HP-UX 9.0, rendering of Asian character data requires that two fonts be opened to correctly render all Asian characters. When an Asian font is opened with `XLoadFont()` or `XLoadQueryFont()`, the font is checked for the existence of an associate font property. If this property exists, an additional font is opened and is transparent to the application. All calls to calculate metrics or to perform rendering with the font causes the implementation to also use the associate font when present.

With support for 3-byte EUC and 4-byte EUC with HP-UX 10.0, rendering Asian character data often requires that more than two fonts be opened and used. The old associate font mechanism could not be reliably expanded to handle these cases. In addition, X11R5 and Motif 1.2 provide alternate, standard mechanisms for rendering internationalized text data. Because of this, support for the associate font mechanism will be removed in a future release of HP-UX.

Applications directly calling the X library to render international text should replace their calculation of font metrics and their calls to `XDraw*`, `XLoadFont()`, and `XLoadQueryFont()` with the following:

<code>XCreateFontSet()</code>	
<code>XExtentsOffFontSet()</code>	
<code>XmbDrawString()</code>	<code>XwcDrawString()</code>
<code>XmbDrawImageString()</code>	<code>XwcDrawImageString()</code>
<code>XmbDrawText()</code>	<code>XwcDrawText()</code>
<code>XmbTextEscapement()</code>	<code>XwcTextEscapement()</code>
<code>XmbTextExtents()</code>	<code>XwcTextExtents()</code>
<code>XmbTextPerCharExtents()</code>	<code>XwcTextPerCharExtents()</code>

Applications and users who set Motif `FontList` resources and desire rendering of internationalized text data should modify those resources to specify font sets. As of Motif 1.2, a `FontList` resource can contain any combination of font structs and font sets. A font set is specified in a `FontList` resource by adding a “:” to the end of the `FontList` entry. For a font set, a `FontList` entry can be one or

more “;” separated base names. A base name is either a font name (such as `jpn.8x18`) or an X Logical Font Description (XLFD). A typical `FontList` setting for Japanese might be `app_name*Text.fontList: *-mincho-*-18-*`:

The “:” character at the end of the `FontList` resource specification is mandatory to use font set technology.

To cause Motif to use font sets correct in the application, either `setlocale()` or `XtSetLanguageProc()` must be called prior to the toolkit being initialized. As of 10.0, HP VUE was converted to use X11R5, Motif 1.2, and the new font set technology. So font resources for all HP VUE clients on 10.x should be set to use font sets if internationalized functionality is desired.

## Terminal Emulators

`dtterm` is a new terminal emulator that is shipping with HP-UX 10.x. `dtterm`:

- Implements the ANSI standard terminal escape sequences as they are implemented by the VT220 terminal and adds the ISO standard color escape sequences.
- Supports all 8-bit locales as well as the Asian EUC locales.
- Is tightly integrated with the VUE environment, providing online help as well as saving user setting across VUE sessions.

For more information on `dtterm`, refer to the `dtterm(1X)` manpage. `hpterm` has changed in several ways for this release:

- `hpterm` is Motif 1.22 and X11R5 compliant (as is the rest of VUE). You no longer need to install the `hpterm` contained in `/usr/bin/X11/hpterm.R5.sh.Z` to access the Motif 1.2/X11R5 features.
- `hpterm` now supports Motif 1.2 font lists. This gives you greater flexibility in specifying fonts for `hpterm` and makes `hpterm` more Motif compliant. The result of this change is that the `hpterm` font (class `FontStruct`) resource has been renamed the `userFont` (class `FontList`) resource. See the `hpterm(1X)` manpage for more information.
- The ability to dynamically change `hpterm`'s keyboard language has been removed. This functionality required proprietary technology and has become impossible to implement across all HP platforms. See “Changes in Keyboard Functionality” earlier in this section for more information. `hpterm` now

## 5-104 Other Operating System and Subsystem Changes

depends on the user to properly initialize the *LANG* environment variable to operate correctly in the desired locale.

- **hpterm** does not support 3- and 4-byte locales; 3- and 4-byte characters are silently discarded from the data stream. If you need a terminal emulator that fully supports locales, you should use **dtterm**.

### **Font Server: Dynamically Loaded Rasterizers**

The 10.x X11 font server dynamically loads its scalable font rasterizers. The **rasterizers =** line in the `/etc/X11/fs/config` file, as installed, loads all scalable font rasterizers shipped by HP. If you are using the font server with an alternate configuration file, you should copy this line into that file. This line is subject to change because new rasterizers will be introduced in future releases.

---

## **X Windows**

### **X11 Fonts**

*For 10.20:*

Wildcard aliases for X Window fonts provide the ability to use the wildcard character in the names of fonts that will be aliased. This can be done by putting the line

```
WILDCARD_ALIASES_SUPPORTED
```

in a `fonts.alias` file in a font directory. With this line, any '\*' in a font name is treated as a wild card when matching. If it does match, the alias is used in place of the specified font.

The old Intellifont rasterizer has been added. This solves a problem with the quality of Japanese Intellifont fonts.

Wildcard aliases will allow system administrators, or providers of scalable fonts, to specify bitmap fonts that will replace scalable fonts at smaller font sizes. This is most important for Asian fonts because scalable Asian fonts become unreadable at smaller font sizes.

## **Motif, Xlib, and Intrinsic Libraries**

*For 10.20:*

X11R6 libraries are shipped with X11R5 libraries. Two Motif libraries (versions R5 and R6) are shipped.

The default build environment for Motif is Version R6. If you want to remain with Version R5, you must modify `makefile`.

Forward compatibility (from X11R5 to X11R6) is guaranteed. However, X11R6 libraries are not backward compatible. Generally, X11R6 is faster than X11R5 in most cases.

## **X11 Clients**

*For 10.20:*

### **Color Support for X11 Clients**

`Xcms.txt` is shipped as part of the X Consortium standard filesets. This file causes the `Xcms` color scheme to be used and thus might result in slightly different image colors. HP did not ship this file in earlier releases, so for consistency, `Xcms.txt` is shipped in this release with the contents commented out. If you want to use the `Xcms` color scheme provided in `Xcms.txt`, you must uncomment the contents of this file.

### **Xterm**

HP-UX 10.20 includes both the X11 Release 5 and Release 6 versions of `xterm`, delivered in the following directories as unsupported software:

- R5: `/usr/bin/X11/xterm`
- R6: `/usr/contrib/bin/X11/xterm`

The functionality of `xterm` is superseded by the industry-standard `dtterm` terminal emulator, making `xterm` redundant. The Release 5 version of `xterm` will not be shipped with the next release of HP-UX; it is provided with this release to facilitate the transition from `xterm` to `dtterm`.

## **5-106 Other Operating System and Subsystem Changes**



## **X11 Server and X11 Window System**

*For 10.10:*

The following topics describe changes in the 10.10 release of the X11 Server compared to the 9.x release of the X11 Server.

- R6-based X server—*new for 10.10*
- HP X server information file.
- X\*screens file.
- Supported Graphics Devices
- Double Buffer Extension (DBE)—*new for 10.10*
- Multi-head support
  - Multi-screen support
  - Single Logical Screen (SLS)—*new for 10.10*
- Shared Memory Extension (MIT-SHM)—*new for 10.10*
- Shared Memory Transport (SMT)—*new for 10.10*
- X Image Extension (XIE)—*new for 10.10*
- Display Power Management Signaling—*new for 10.10*
- New graphics support
- Dynamic loading of DDX drivers and extensions
- Graphics in SAM—*new for 10.10*
- Include Inferiors Fix—*new for 10.10*

Detailed X server information is available on most of the above topics in the ASCII file `/usr/lib/X11/Xserver/info/screens/hp`.

### **R6-Based X11 Server**

*For 10.10:*

The 10.10 X11 server is based on the X Consortium's X11 Revision 6 sample implementation, augmented with HP modifications. The most important additions to the Consortium's R6 X11 server were already included in HP's version of the R5-based server, including recycling of resource ID's and the double buffer extension. The newer R6 server contains a number of bug fixes, support for additional extensions (see below), and support for new display hardware (see below).

Although the 10.10 server is based on the R6 sample, it has been built with the R5 version of the X11 library. A full R6 X11 implementation will be available in a later release.

## **HP X Server Information File**

*For 10.10:*

A new information file is located in `/usr/lib/X11/Xserver/info/screens/hp`. It describes features unique of HP's X server, provides information on how to configure the X server, and includes a list of supported X configurations. For each supported graphics device, device-dependent configuration information is provided.

The following topics are covered:

- General Information
- X\*screens File
- Miscellaneous Topics
  - Double Buffer Extension (DBE)
  - Multi-Buffering Extension (MBX)
  - Single Logical Screen (SLS) support
  - Shared Memory Transport (SMT)
  - HP Color Recovery
  - Dynamic loading of drivers and extensions
  - Shared memory usage
  - Other topics
- Supported X Configurations
- Device-dependent information for these display types:
  - Integrated Color Graphics
  - Entry (Internal) Color and Grayscale Graphics
  - CRX24[Z] Graphics
  - CRX48Z Graphics
  - HCRX Series Graphics
  - HP VISUALIZE Series Graphics
  - Freedom Series Graphics
  - VRX Graphics

## **5-108 Other Operating System and Subsystem Changes**

## **X\*screens Format**

### **New X\*screens File Format.**

*For 10.10:*

The format of the `/etc/X11/X*screens` file has changed. A tool is available to convert an X\*screens file in the old format to the new format. The tool is located in `/usr/bin/X11/convertscr`. Type the command `/usr/bin/X11/convertscr -h` to learn how to use the tool. For a detailed description of the new format, see the `/etc/X11/X*screens` file.

Many of the X\*screens options are device dependent. To find out what X\*screens options are supported on your display, look in the appropriate information file for more details (for example, `/usr/lib/X11/Xserver/info/screens/hp`). For more information about configuring the X\*screens file, refer to the section “Graphics in SAM” in this chapter.

**Setting the Default Visual.** The X\*screens file allows you to specify the default visual. Although it is possible to specify any supported visual as the default, it is recommended that you let the X server select the default in order to get the best performance on your display. (To get a list of visuals supported on your display, use the `xdpyinfo` command.) For more information on `xdpyinfo`, refer to the `xdpyinfo(1)` manpage in `/usr/contrib/man/man1.Z`.

When you specify a default visual in the X\*screens file, the first matching visual in the visual list will be chosen. For example, if you specify the following:

```
Default Visual
Depth 8
```

the first depth 8 visual encountered in the visual list will be selected as the default.

If there is any error in the specification of the default visual in the X\*screens file, the X server will select the specified device’s standard default visual.

**Setting the Monitor Size.** The `MonitorSize` option of the X\*screens file can be used to specify the diagonal length of your monitor in units of either inches or millimeters. You only need to set this option if your application requires this

information (for example, in order to compute a pixels-per-inch figure) and your monitor is not the standard 19-inch monitor.

**X Server Environment Variables.** The functionality of most previously supported environment variables is now supported via `ServerOptions` or `ScreenOptions` in the `X*screens` file.

In previous HP-UX releases, environment variables have typically been specified in the `Vuelogin*environment:` line of the `/usr/vue/config/Xconfig` file. You should replace environment variables specified in `Xconfig` with entries in the `X*screens` file.

---

**Note**            The 10.01 release of the X server checked for these obsolete environment variables. The 10.10 version does not make this check.

---

Those environment variables directly replaced with the new `X*screens` format are listed below.

Old environment variable	New syntax
GRM_SIZE	GraphicsSharedMemorySize
WMSHMSC	GraphicsSharedMemorySize
HP_COUNT_TRANSPARENT_IN_OVERLAY_VISUAL	CountTransparentInOverlayVisual
CRX24_COUNT_TRANSPARENT_IN_OVERLAY_VISUAL	CountTransparentInOverlayVisual
HP_DISABLE_COLOR_RECOVERY	DisableColorRecovery
HP_ENABLE_OVERLAY_TRANSPARENCY	EnableOverlayTransparency
HPGCRX_IMAGETEXT_VIA_BITMAP	ImageTextViaBitMap
MBX_SWAP_BUFFERS_ON_VBLANK	MbxSwapBuffersOnVBlank
HP_3_BIT_CENTERCOLOR	3BitCenterColor
CRX24_FULL_DEFAULT_VISUAL	DefaultVisual Class <value>
ES_VIDEO_FORMAT	FreedomVideoFormat

Here is an example:

Old environment variable:

```
export HPGCRX_IMAGETEXT_VIA_BITMAP=TRUE
```

New screen option:

```
ScreenOptions
ImageTextViaBitMap
```

## 5-110 Other Operating System and Subsystem Changes

Remember that ServerOptions will affect all screens in a multi-screen configuration.

## Supported Graphics Devices

*For 10.10:*

The following table summarizes the graphics devices that are supported on each of the HP 9000 Series systems.

HP 9000 Supported Graphics Devices Table

Graphics Devices	Graphics Product #	Supported HP 9000 Models
Integrated Grayscale Graphics *	A1991A Grayscale	712 (all models), 715/64, 715/80, 715/100, 725/100
Integrated Color Graphics with HP Color Recovery **	HPA208LC	712 (all models), 715/64, 715/80, 715/100, 725/100, 748i/64, 748i/100, V743/64, V743/100
Color Graphics card with HP Color Recovery	A4077A A2999A	712 (all models) 715/64, 715/80, 715/100, 725/100, 748i/64, 748i/100, J200, J210, K100, K200, K210, K400, K410, D310, D350
Dual Color Graphics card	A4078A	715/64, 715/80, 715/100, 725/100, J200, J210
Entry/Internal Grayscale Graphics *	M/A	705, 710, 715/33, 715/50, 715/75, 725/50, 725/75
Entry/Internal Color Graphics **	A208LC	705, 710, 715/33, 715/50, 715/75, 725/50, 725/75, 745i/50, 745i/100, 747i/50, 747i/100
CRX Dual CRX	A1659A A2262A	720, 730, 735, 735/125, 750, 755, 755/125, 747i/50, 747i/100,
GRX	A4053A	720, 730, 735, 735/125
CRX-24	A2673A A2271A A2272A	715/33, 715/50, 715/75, 720, 725/50, 725/75, 730, 735, 735/125, 747i/50, 747i/100, 750, 755, 755/125

CRX-24Z	A2674A	715/33, 715/50, 715/75, 720,
	A1454A	725/50, 725/75, 730, 735, 735/125,
		750, 755, 755/125
-----+		
CRX-48Z	A2675A	715/50, 715/75, 715/100, 725/50,
		725/75, 725/100, 735, 735/125,
		755, 755/125, J200, J210
-----+		
HCRX-8	A4070A	715/64, 715/80, 715/100, 725/100,
HCRX-8Z	A4079A	J200, J210
-----+		
HCRX-24	A4071A	715/64, 715/80, 715/100, 725/100,
HCRX-24Z	A4179A	J200, J210
VISUALIZE-8	A4441A	
VISUALIZE-24	A4442A	
-----+		
VISUALIZE-48	A4244A	J200, J210
-----+		
Freedom Series	A4091A	715/80, 715/100, J200, J210
Graphics	A4093A	
-----+		
PersonalVRX		720, 730, 735, 735/125, 750, 755,
TurboVRX		755/125
(No longer on the Corporate Price List)		
-----		

- \* Integrated Grayscale Graphics and Internal Grayscale Graphics is supported on high resolution (1280x1024) for all models specified above.
- \*\* Integrated Color Graphics and Internal Color Graphics is supported on both medium resolution (1024x768) and high resolution (1280x1024) configurations of the Series 700 Models 705, 710, 712 (all models), and 715/33. High resolution is supported on all other models specified above.

M/A: Available only as an option on mature platforms.

## Double Buffer Extension (DBE)

*For 10.10:*

DBE is an extension to the X server that provides a double-buffering Application Programming Interface (API). Note that MBX (the Multi-Buffering eXtension to X) has not been adopted as an industry standard, as DBE has. Thus, it is recommended that applications that use MBX be ported to DBE. For more information about DBE and the API, consult the

## 5-112 Other Operating System and Subsystem Changes

`/usr/lib/X11/Xserver/ info/screens/hp` information file or these DBE manpages:

```
DBE
XdbeQueryExtension
XdbeAllocateBackBufferName
XdbeBeginIdiom
XdbeDeallocateBackBufferName
XdbeEndIdiom
XdbeFreeVisualInfo
XdbeGetBackBufferAttributes
XdbeGetVisualInfo
XdbeSwapBuffers
```

### **Multi-Head Support**

*For 10.10:*

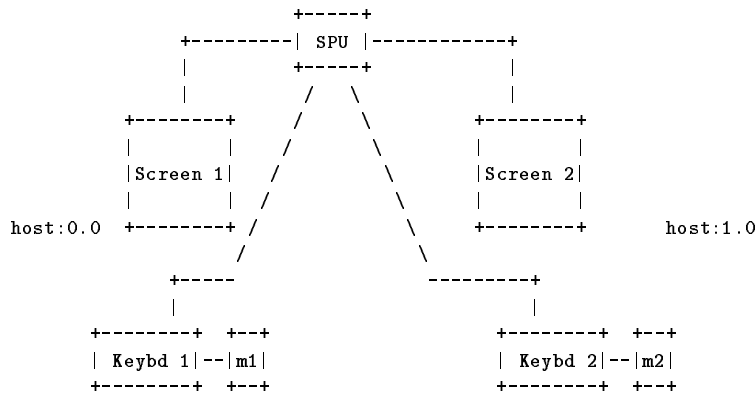
The following definitions are included to help reduce confusion between the terms multi-head, multi-screen, multi-seat, and single logical screen (SLS).

- **Multi-Head**—A configuration with multiple graphics devices (heads) used concurrently. Any multi-screen or multi-seat configuration is referred to as a multi-head configuration.
- **Multi-Screen**—A configuration in which a single X server with a mouse and keyboard drives multiple graphics devices (where each head is a different X11 screen) concurrently while only allowing the cursor, not windows, to be moved between heads.
- **Multi-Seat**—A configuration with multiple instantiations of the X server, each with its own mouse, keyboard, and heads. Multi-seat is not supported on any HP-UX 10.x release.
- **Single Logical Screen (SLS)**—A configuration in which a single X server with a single mouse and keyboard drives multiple homogeneous graphics devices (heads) concurrently while allowing the heads to emulate a large single screen. This differs from a multi-screen environment by allowing windows to be moved and displayed across heads. See the section in this document on Single Logical Screen for more information

Different monitor resolutions are not supported with multi-head configurations unless stated otherwise in the table.

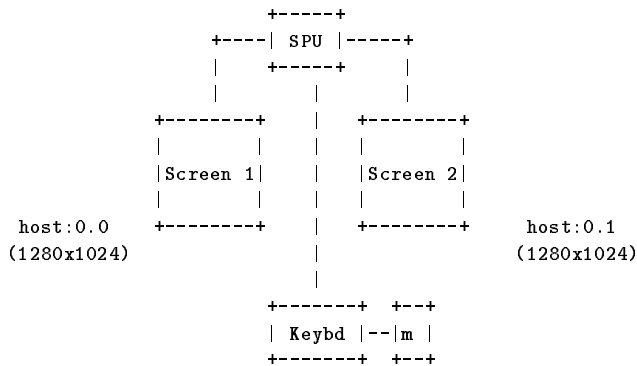
The following diagrams show examples of the three different types of multi-head configuration.

### Multi-seat example

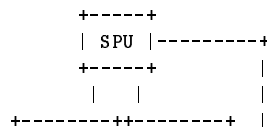


Note: multi-seat is not supported in any HP-UX 10.\* release.

### Multi-screen example



### Single Logical Screen example





```

      |           ||           | |
      |Graphics||Graphics| |
      |Device 1||Device 2| |
      |           ||           | |
      +-----+-----+ |
      |
      +-----+
      | Keybd |--|m |
      +-----+ +---+

```

host:0.0  
(2560x1024)

## Multi-Screen Support

*For 10.10:*

Running one X server on more than one graphics display is referred to as a “multi-screen” operation. The keyboard and pointer are shared among the screens. The screens are defined in the X\*screens file by specifying the appropriate special device files. Windows can be placed on any of the screens and the cursor can move between the screens, but it is not possible to move windows from one screen to another, or have windows span more than one screen.

See the `/usr/lib/X11/Xserver/info/screens/hp` file for information on special device files and `/usr/lib/X11/X0screens` for additional information.

## Single Logical Screen (SLS)

*For 10.10:*

SLS is a mechanism for treating homogeneous multi-head configurations as a single “logical” screen. This allows the moving/spanning of windows across multiple physical screens. The word “homogeneous” is included because SLS only works if the graphics devices included in the “SLS Configuration” are of the same type. See the `/usr/lib/X11/Xserver/info/screens/hp` file for information on supported configurations and enabling SLS functionality.

There are some considerations when deciding to configure for multi-screen or SLS configurations:

- 3D applications, such as Starbase- and PEX-based clients, should use the multi-screen configuration. These clients need direct hardware access (DHA) for good performance and DHA is not available in SLS configurations.

- 2D applications that need to display images that are larger than a standard (1280x1024 pixels) screen can take advantage of the larger effective screen size of an SLS configuration. For example, a 2x2 SLS configuration is like having a 2560x2048 pixel screen.
- Both the VUE and CDE desktops place various dialog and information boxes in the center of the screen. In some SLS configurations, this will result in the box being split between two or more screens, which may negatively impact readability.

### **Shared Memory Extension (MIT-SMT)**

*For 10.10:*

The MIT shared memory extension provides both shared memory XImages and shared memory pixmaps based on the SYSV shared memory primitives.

Shared memory XImages are essentially a version of the XImage interface where the actual image data is stored in a shared memory segment and thus need not be moved through the Xlib interprocess communication channel. For large images, using this facility can result in performance increases.

Shared memory pixmaps are a similar concept implemented for the pixmap interface. Shared memory pixmaps are two-dimensional arrays of pixels in a format specified by the X server, where the pixmap data is stored in the shared memory segment. In all other respects, shared memory pixmaps behave the same as ordinary pixmaps and can be modified by the usual Xlib routines. In addition, it is possible to change the contents of these pixmaps directly without the use of Xlib routines merely by modifying the pixmap data.

**Supported Devices.** The X server supports the MIT shared memory extension on the following devices:

- Entry (Internal) Color Graphics
- Integrated Color Graphics
- CRX-24[Z]
- CRX-48Z
- HCRX-8[Z]
- HCRX-24[Z]
- VISUALIZE-8
- VISUALIZE-24
- VISUALIZE-48

### **5-116 Other Operating System and Subsystem Changes**

## Shared Memory Transport (SMT)

*For 10.10:*

The 10.10 X server supports a new implementation of Shared Memory Transport (SMT) offering improved performance characteristics under some circumstances. The new implementation offers advantages to certain applications and installations, but disadvantages to others.

Hewlett-Packard has observed that there are cases where use of SMT might degrade system performance, such as VUE login time (from login to first terminal window). The conditions under which SMT will show improvements or degradations in performance are complex. Therefore, HP recommends you experiment to determine if SMT is of benefit for your application and that the “extra files” discussed below are not created. HP is investigating ways to address these issues without impacting other areas of system performance. See “Possible Extra Files Generated with SMT” for other concerns.

Shared Memory Transport is a means to transport large amounts of data from the client to the server when both are running on the same system. It is distinct from the MIT Shared Memory Extension, which is specifically for various types of images, although SMT can be used with that extension. SMT offers performance advantages for operations that move large amounts of data in a single request, such as polyline or polypoint, and for images when the MIT Shared Memory Extension is not used. SMT will work with the Big Requests Extension, but any performance benefit depends on the size of the actual extended size request. There are some X requests for which no improvement is expected under SMT and general desktop GUI performance shows some slowdown when SMT is enabled.

With the 10.10 release, the SMT is an optional transport that can be enabled by setting an environment variable, `XALLOW_SMT`, in the X server’s environment (see below for how to do this for VUE or CDE). The X11 Server will check this variable on startup. If it is not defined (the default state), all local client-server communication will be via Unix Domain Sockets (UDS), which was the only local transport on recent releases prior to 10.10. (Remote client-server communication will be via Internet Sockets as usual).

If the `XALLOW_SMT` environment variable is defined when the X server is started, the SMT will be the transport whenever a hostname of any of the forms listed below are used and when the client and server are actually on the same host.

Note that “:0.0” is used for simplicity. This behavior is equally applicable for displays such as “:1.0”.

```
:0.0
local:0.0
hostname:0.0
shmlink:0.0
```

A hostname of the form

```
unix:0.0
```

will force the use of Unix Domain Sockets (UDS), as stated above.

A hostname of the form

```
nn.nn.nn.nn:0.0
```

where *nn.nn.nn.nn* is an IP address, will force the use of Internet Sockets, which is the remote transport normally used even when the client is running locally. This will be slower.

It is possible that an application that violates the X interface standard will run correctly using UDS, but hang or core dump when using SMT. If this happens, set your DISPLAY environment variable as DISPLAY=unix:0 or specify unix:0 using the command line -display parameter to run the application compatibly without the performance change of SMT.

For example, use one of the following to start `xclock` on screen 0 with UDS (ksh syntax):

```
$ DISPLAY=unix:0 xclock
```

or

```
$ xclock -display unix:0
```

If you choose to enable the SMT, see the information file `/usr/lib/X11/Xserver/info/screens/hp` for additional details, in particular the note to programmers.

## 5-118 Other Operating System and Subsystem Changes

**Starting VUE or CDE with the SMT Transport.** If you are running applications that move large amounts of data in single operations, the Shared Memory Transport may offer an advantage. You can run VUE or CDE with the SMT as the default transport by making changes in the VUE or CDE Xconfig file.

To run VUE or CDE with SMT as the default transport, change the file `/usr/vue/config/Xconfig` (VUE) or `/usr/dt/config/Xconfig` (CDE). For `/usr/vue/config/Xconfig`, remove the command and edit the line where the `Vuelogin*environment` resource is set:

```
Vuelogin*environment: XALLOW_SMT=Defined
```

The change to `/usr/dt/config/Xconfig` is similar:

```
Dtlogin*environment: XALLOW_SMT=Defined
```

After making the change, shut down and restart VUE or CDE. The result will be that SMT will be the local transport mechanism for the desktop clients and any clients that use the SMT forms of the display names listed above.

**Possible Extra Files Generated with SMT.** You may see files created in the `/var/spool/sockets/X11` directory under certain circumstances when SMT is the transport mechanism. There might be a large number of such files and they might consume significant space. Because of this, HP does not recommend use of SMT on diskless nodes at this time.

If the following is true:

- your system is using NFS diskless (or `/var/spool/sockets/X11` is on a NFS file system)
- the X client application is started using SMT as described above
- the X client is run by root or is `set-user-id` to root
- the X server is not running as root (which is normally the case)

the message “Server Shared Mem truncate failed” might be printed by the X server and files with names of the form `,X,*` or `.nfsXXXX` might be left in `/var/spool/sockets/X11`.

The application will run normally and no damage is done to the X server. The files with names prefixed by `,X,` can be removed at any time. The `.nfsXXXX` files can be removed when the X server is not running. The files of the form

.nfsXXXX can also be removed if they were created before the most recent X server startup. *DO NOT* remove any such files that are contemporary with the current X server.

### **X Image Extension (XIE)**

*For 10.10:*

The 10.10 X Server is 'pre-enabled' with the X Image Extension (XIE). This means that the X server comes with a dynamically loadable implementation of the sample XIE. Support for the development of X clients that use the XIE is not part of the 10.10 release, but existing clients that use the XIE will be able to run on the 10.10 X server. Also, users who wish to implement the sample XIE client-side library (available in the R6 distribution from the X Consortium) will be able to develop XIE-clients and run them on the 10.10 X server.

### **Display Power Management Signaling**

*For 10.10:*

Monitors constitute a large percentage of the power used by a workstation even when not actively in use (such as during screen blanking). To reduce the power consumption, the Video Electronic Standards Association (VESA) has defined a Display Power Management Signaling (DPMS) standard that can be used to greatly reduce the amount of power being used by a monitor during screen blanking.

The X server features the ability to make use of DPMS on the following graphics devices:

- HCRX-8[Z]
- HCRX-24[Z]
- VISUALIZE-8
- VISUALIZE-24
- VISUALIZE-48

The following table is a description of the states that are defined by VESA. The Power Savings column indicates (roughly) the level of power savings achieved in the given state. The Recovery Time is the amount of time that the screen takes to return to a usable state when the screen saver is turned off (by pressing a key or the moving the mouse).

## **5-120 Other Operating System and Subsystem Changes**

Screen Option Value	State	DPMS Compliance Requirements	Power Savings	Recovery Time
0	Screen Saver	Not Applicable	None	Very Short (< 1 sec)
1	Stand-by	Optional	Minimal	Short
2	Suspend	Mandatory	Substantial	Longer
3	Off	Mandatory	Maximum	System Dependent

The actual amount of power saved and the recovery time for each of the states is monitor-dependent and can widely vary. You can compensate for this by choosing an appropriate level for the monitor that is currently in use.

By default, the DPMS level used is the Screen Saver (that is, no power savings). If you want to use power saving during screen blanking, set the following X\*screens file Screen Option before starting the server:

```
MinimumMonitorPowerSaveLevel #
```

where # is replaced with the single digit 0, 1, 2, or 3 as specified in the Screen Option Value column in the above table.

### **New Graphics Support**

*For 10.10:*

Grayscale is supported on the “Integrated Color Graphics” device. The current grayscale monitor is A2088A (a 1280 x 1024, 19-inch monitor, with a 70-Hz refresh rate). 3D libraries (such as Starbase, PEX, and PHIGS) are not

supported with this grayscale. (Grayscale support on the “Integrated/Entry Color Graphics” devices does support 3D libraries.)

New graphics supported in 10.10 include the HP VISUALIZE Series Graphics and the Freedom Series Graphics. See the information file `/usr/lib/X11/Xserver/info/screens/hp` for more information.

A screen option “FreedomVideoFormat” replaces the HP-UX 9.x environment variable `ES_VIDEO_FORMAT`. Freedom Series graphics devices can support several different video formats. The appropriate video format must be selected to support the specific display device connected to the Freedom accelerator. See the `/etc/X11/Xserver/info/screens/hp` file for details.

### **New Dynamic Loading of DDX Drivers and Extensions**

*For 10.10:*

HP’s R6 server now dynamically loads all device drivers and some extension modules at startup time. This feature is transparent to X server users.

---

**Caution**      Altering or removing files under `/usr/lib/X11/Xserver` may prevent the X server from running.

---

### **Graphics In SAM**

*For 10.10:*

Starting with HP-UX 10.10, system administrators will be able to configure the `X*screens` file via HP’s system administrator’s tool, SAM. Manual configuration of the file is still supported, but where possible, you are encouraged to begin configuring graphics with SAM.

### **Include Inferiors Fix**

*For 10.10:*

When a client creates an X Graphics Context (GC), it is possible to specify the `subWindowMode` component. The two possible values are `ClipByChildren` (default) and `IncludeInferiors`. If the GC specifies `ClipByChildren`, any rendering to a window with inferior windows (child, grandchild, and so on) will appear only in the destination window. In other words, the rendering will not

## **5-122 Other Operating System and Subsystem Changes**



take place inside the inferiors. If the GC specifies `IncludeInferiors` and the same rendering request is made, it is the responsibility of the X Server to ensure that the rendering is *not* clipped from the inferior windows. In other words, the rendering will appear in the destination window as well as all inferior windows.

With the advent of multi-layer devices, the `IncludeInferiors` mode became defective. Depending upon which layer or hardware buffer the destination drawable and inferior windows were in, the rendering may or may not take place. What is worse, the `GetImage` protocol clearly specifies that the default `GetImage` behavior is to include the contents of inferior windows.

With the 10.10 release, HP offers a solution to the `IncludeInferiors` defect. Some users create their test image archives using `XGetImage` (which currently returns incorrect data for multi-layer devices). Therefore, the `Include Inferiors Fix` will not be enabled by default. To enable the `Include Inferiors Fix`, add the following `Screen Option` to the `X*screens` file:

`EnableIncludeInferiorsFix`

This gives a system administrator control over when the fix is active and when it is not. This way, each user can evaluate whether it is beneficial to enable this fix or not.

The `EnableIncludeInferiorsFix` `Screen Option` will have no effect on `VISUALIZE-48` displays with this release. This will be corrected in a future release.



## Networking

---

This chapter covers the following topics:

- BSD Sockets—*changed at 10.10*
- **Data Link Provider Interface (DLPI)**—*changed at 10.20*
- **FDDI SNMP Subagent**—*changed at 10.20*
- **Internet Services (ARPA Services)**—*changed at 10.20 and 10.10*
  - **4-Byte EUC rlogin**—*changed at 10.20*
  - **Secure Internet Services**—*changed at 10.20*
  - **sendmail for 10.20**—*changed at 10.20*
  - **sendmail for 10.0**—*changed at 10.0*
- LAN/9000—*changed at 10.10*
- LAN Cable Disconnect
- NETTRACELOG
- **Networking \_r Routines**—*changed at 10.20*
- NFS
- NIS—*changed at 10.20*
- NS/9000 and vt3k/9000
- **ONC/NFS Large UID Support**—*changed at 10.20*
- PMTU
- SLIP and CSLIP
- Sockets—*changed at 10.10*
- Streams—*changed at 10.10*
- TERMIO
- XTI—*changed at 10.10*

---

## BSD Sockets

*For 10.10:*

HP-UX 10.10 supports two sets of sockets behavior: HP-UX 10.01 and UNIX95 (X/Open Sockets). By default, you receive HP-UX 10.01 sockets. To use UNIX95, you must make an addition to your `make` scripts by including the `-l xnet` argument with the `c89` or `cc` utilities.

You can also use the `lint` utility to check for possible violations to the X/Open Sockets specification by including `-l xnet` to the argument list of the `lint` utility.

You must also define the following for UNIX95 sockets behavior:

```
XOPEN_SOURCE_EXTENDED=1
```

Refer to the *BSD Sockets Interface Programmers Guide* for a list of supported calls for these two types of sockets behavior. The manual summarizes differences between the two sets of sockets behavior for each call.

*For 10.0:*

Two new socket options have been added at 10.0 for more flexibility in TCP/IP connections and to conform to RFC 1122, *Requirements for Internet Hosts*. The new IP socket option, `IP_TOS`, allows applications to set the IP Type-Of-Service flags, and may affect the route chosen for IP packets over routers or gateways that implement the IP type of service. The new TCP socket option, `TCP_MAXRETRANS`, allows applications to close a connection before the system-wide maximum number of allowable retransmissions (12) is reached. This can improve the behavior of applications in fast, reliable networks where multiple retransmissions are rare.

For more information, refer to the *Berkeley IPC Programmer's Guide* and the `ip(7p)` and `tcp(7p)` manpages.

---

## Data Link Provider Interface (DLPI)

*For 10.20:*

The definitions for the three levels of promiscuous mode have been modified to be consistent with other DLPI providers' definitions. The definitions are:

■ **DL\_PROMISC\_PHYS**—

Prior to a `DL_BIND_REQ`, the DLPI user receives all traffic on the wire regardless of SAP or address. After a `DL_BIND_REQ`, the DLPI user receives all traffic on the wire that matches one of the protocol(s) the user has bound to on the promiscuous stream (this includes protocols bound to with the `DL_SUBS_BIND_REQ`).

■ **DL\_PROMISC\_SAP**—

Prior to a `DL_BIND_REQ`, the DLPI user receives all traffic originally destined for the interface (physical addresses, broadcast addresses or bound multicast addresses) and that matches *any* SAP enabled on that interface. After a `DL_BIND_REQ`, the DLPI user receives only those packets originally destined for the interface that matches one of the protocol(s) bound to on the promiscuous stream.

■ **DL\_PROMISC\_MULTI**—

Prior to a `DL_BIND_REQ`, the DLPI user receives all multicast packets on the wire regardless of SAP. After a `DL_BIND_REQ`, the DLPI user receives all multicast packets that matches one of the protocol(s) bound to on the promiscuous stream.

*For 10.0:*

The Data Link Provider Interface (DLPI) is part of the 10.x HP-UX operating system. The DLPI provides STREAMS-based link-level access to network interfaces. The HP-UX version of DLPI currently supports the following network interfaces: Ethernet/IEEE 802.3, IEEE 802.5, FDDI, and Fibre Channel.

For more information, refer to the *Data Link Provider Interface* manual (98194-90054).

---

## FDDI SNMP Subagent

*For 10.20:*

The NIO FDDI product will include an FDDI SNMP Subagent that provides RFC 1512-defined network management functionality to work within the OpenView Network Management framework. This subagent runs as a daemon process to GET and SET various SMT 7.2 statistics, as defined in RFC 1512.

The lan6 NIO FDDI driver is enhanced to handle SMT GET and SET requests from DLPI. A user-space daemon has been developed to act as the subagent between the SNMP Master Agent and the FDDI driver. Further, startup and shutdown scripts have been written to start and stop this daemon.

### Impact

This feature now allows OpenView users to manage their FDDI nodes as described in RFC 1512.

### Restrictions

This feature is only available on the NIO FDDI (lan6) product, which means that this will only run on platforms that have an NIO bus with FDDI installed. The available platforms are: HP 9000/800 E, G, H, I, K, and T models.

---

## Internet Services (ARPA Services)

*For 10.20:*

The Internet Services product is essential for networking with the UNIX core software. These services, described in this section, are:

- 4-Byte EUC rlogin
- Secure Internet Services
- sendmail

*For 10.10:*

- mrouted

- Name Service Switch
- Talk Service (`talk/ntalkd`)
- `gated`

*For 10.0:*

Internet Services was the new 10.0 name for what has been called ARPA Services in pre-10.0 HP-UX releases. The Internet Services include not only the ARPA Services, but also services from Berkeley (BSD) and other sources.

The following product areas are included:

ARPA Services	Berkeley Services	Routing	Booting	Other
Telnet, FTP, SMTP	r-services, <code>sendmail</code> , BIND (DNS), network libraries, <code>finger</code>	<code>gated</code>	<code>bootp</code> , <code>tftp</code> , <code>rbootd</code>	NTP, <code>ddfa</code> , <code>mailx</code> , <code>elm</code>

The following sections summarize the major changes to this product.

#### 4-Byte EUC rlogin

*For 10.20:*

The `rlogin/rlogind` internet service has been enhanced to work with 4-byte EUC (Extended Universal Character set).

To support this feature, the pseudo terminal in `rlogind` now uses a STREAMS-based `pty` driver. The STREAMS master `pty` driver is `ptm` and the device used is `/dev/ptmx`. The STREAMS slave `pty` driver is `pts` and the slave devices used are `/dev/pts/number`.

This change to `rlogin` has an important effect on one of the tuning parameters of the kernel, namely `NSTRPTY`. Because `rlogind` now uses the STREAMS-based pseudo terminal, the kernel parameter to be tuned for `rlogin` pseudo terminals is `NSTRPTY`, rather than `NPTY`, which is used for pseudo terminals using the traditional `pty` driver.

An additional effect of this change is that for `rlogin`-based pseudo terminals, an execution of the `tty` command will now display `/dev/pts/number` rather than `/dev/ttynumber`.

## Secure Internet Services

*For 10.20:*

In HP-UX 10.20 there is a new optionally installable product called InternetSvcSec. This product provides alternative versions of the following internet services: `ftp`, `remsh`, `rcp`, `rlogin`, and `telnet`. These alternative versions of the services incorporate Kerberos Version 5 Beta 4 authentication and authorization and are referred to as the “Secure Internet Services”.

The Secure Internet Services are meant as replacements for their non-secure counterparts. The main benefit of running the Secure Internet Services is that user authorization no longer requires transmitting a password in a readable form over the network. Additionally, when both systems are operating in a Kerberos V5-based secure environment, the Secure Internet Services ensure that a local and remote host are mutually identified to each other in a secure and trusted manner and that the user is authorized to access the remote account.

For `ftp/ftpd`, `rlogin/rlogind`, and `telnet/telnetd`, the Kerberos V5 authentication involves sending encrypted tickets instead of a readable password over the network to verify and identify the user. Although `rcp/remshd` and `remsh/remshd` do not prompt for a password, the secure versions of these services ensure that you are authorized to access the remote account.

---

<b>Note</b>	None of the Secure Internet Services encrypts the session beyond what is necessary to authorize the user or authenticate the service. Thus, these services do not provide integrity-checking or encryption services on the data or on the remote sessions.
-------------	--

---

The Secure Internet Services use Kerberos-specific port entries in `/etc/services` and the entries for `kshell` and `klogin` in `/etc/inetd.conf`. These entries are automatically present in HP-UX 10.20.

### System Requirements

Hardware Requirements: HP 9000 Series 700 or Series 800

Software Requirements: HP-UX 10.20

## 6-6 Networking



Prerequisite Software: HP DCE fileset (DCE-Core.DCE-CORE-RUN)  
Internet Services fileset (InternetSrvcs.INETSVCS-RUN)

Disk Space: This product requires an additional 3.6 Mbytes  
more of disk space.

Memory: No additional memory is required.

## Environment

The Secure Internet Services require a Kerberos network authentication services environment which includes a properly configured Key Distribution Center (KDC). Supported KDCs are the HP DCE security server or any third-party KDC based on Kerberos Version 5 beta 4. A properly configured KDC must be running for the Secure Internet Services to work.

## Installing/Enabling

InternetSvcSec is optionally installable. This product should be selected only if you want Kerberos authentication for the `ftp`, `remsh`, `rcp`, `rlogin` and `telnet` services.

To install the product, invoke `swinstall`. The default view of the software is in the form of bundles. Change the software view to products and select the InternetSvcSec product for installation. The product contains the runtime fileset (`INETSVCS-SEC`), as well as filesets containing the manpages. In addition to the client/daemon man pages there is a new manpage called `sis(5)` that contains information common to all the Secure Internet Services, including warning and error messages.

Within `INETSVCS-SEC`, there is a required startup script called `inetsvcs_sec`. To enable the product, invoke the command

```
/usr/sbin/inetsvcs_sec enable
```

For disabling, either invoke

```
/usr/sbin/inetsvcs_sec disable
```

or `disable` and remove the product using the `swremove` command.

---

**Note**            The InternetSvcSec product will take approximately 3.6 Mbytes of additional disk space. No additional memory is required.

---

## Usage

Before using the Secure Internet Services, you must first obtain credentials from the KDC server. Usually this is done by issuing a `kinit` command. Once these initial credentials are obtained, the Secure Internet Services can be used throughout the time period that the credentials are valid. This period is configurable and is typically eight hours. There are no visible differences between using the secure and non-secure versions of the internet services, except that you are not prompted for a password and there are some new Kerberos-related options available. When you are finished with the session, remove accumulated credentials using the `kdestroy` command.

## Compatibility

Depending on how certain options are used with these services, the Secure Internet Services clients will still be able to access non-secure remote hosts and the daemons will still be able to accept requests from non-secure clients.

If any of the Secure Internet Services are installed in an environment where some of the remote systems on the network are non-secure, you can use the `-P` command line option to bypass Kerberos authentication. However, if accessing the host requires a password, the password will be sent in a readable form over the network.

To protect the integrity of passwords on servers, you can prevent remote users from gaining access in a non-secure manner. For `ftpd` and `telnetd` to prevent access from non-secure clients, these daemons should be invoked with the `-A` option. This option enforces Kerberos authentication. For `remshd` and `rlogind` to prevent access from non-secure clients, the entries for `shell` and `login` in the `/etc/inetd.conf` file should be commented out. For any service, if these steps are taken, the client cannot use the `-P` option to bypass authentication for that service.

## **sendmail for 10.20**

*For 10.20:*

`sendmail` is the mail transport agent daemon that implements the SMTP mail protocol. For 10.20, the `sendmail` transport agent daemon is based on the public domain `sendmail` version 8.7. This is an upgrade from the previous HP-UX version of `sendmail`, which was based on version 5.65. The new version offers the same basic functionality as the previous version, but features significant new functionality and configuration flexibility, including a wide variety of new and extended options.

The new features of `sendmail` version 8.7 are:

### ■ **Performance enhancements:**

- Open SMTP connections are now cached for possible future use, rather than being closed immediately.
- Response time to the RCPT command is faster. Rather than expanding all aliases, the new `sendmail` only checks for the existence of an alias and defers the alias expansion until later.

### ■ **Easier to use and more flexible `sendmail.cf` configuration file:**

- `m4` macros are included with instructions on using these macros to build `sendmail.cf` files.
- The configuration file now features descriptive word options rather than single letter options.
- All timeouts are now configurable for controlling the SMTP exchange.
- A `K` command now permits the use of maps for more flexible rulesets; you can create databases of keyed lookups that are used for rewriting addresses in the rulesets.

### ■ **Improved security features:**

- The `sendmail` restricted shell `smrsh` provides a secure way of restricting which programs can be invoked through the aliases or vacation file when mailing to programs. Only programs linked to a specific `smrsh` directory, `/var/adm/sm.bin`, can be used.

- The new `identd` daemon is used by `sendmail` to try to prevent forged mail by logging the user who initiated the `sendmail` connection.

- Two new utilities are now supported:

- `owners`—contacts the local `identd` and lists the login names of the users with outgoing network connections.

- `idlookup`—identifies the owner of an incoming connection by contacting the `identd` of a remote system.

- **Other features added or modified:**

- If an alias has an associated owner-list name, that alias is used to change the envelope sender address. This causes any downstream errors to be returned to the owner-list name.

- The `mtail` utility lists the last  $n$  lines (the default is 20) of the `/var/adm/syslog/mail.log` file. This is useful for quickly looking at recent mail activity.

- For tracing purposes, IP addresses are now logged in “Received:” lines.

- The new `expand_alias` utility is a recursive version of `praliases` that attempts to expand an alias by recursively contacting remote systems and doing an `EXPN alias` until it cannot go further. This can be useful in expanding mail lists.

- To kill `sendmail`, the new command `killsm` replaces the command line option `-bk`.

- a `sendmail.cw` file is used instead of the `cw` macro to define local host names.

- By default, DNS will be used for host look-ups and the aliases file will be used for alias look-ups. A `service.switch` file can be used to indicate that either files or NIS are requested before DNS. For running in a non-DNS environment, the `$j` macro can be used to define a domain name.

- The aliases file is hashed into `.db` files rather than `.pag` and `.dir` files. The `.db` format is more efficient. This hashing occurs whenever a user runs `newaliases`.

## 6-10 Networking

## Compatibility and Installation

*For 10.20:*

The basic `sendmail` functionality is not changed. There should be little impact on backward compatibility. You can still invoke `sendmail` as a mail user agent or mail transport agent with most of the same options.

HP-UX 10.20 `sendmail` uses Version 6 `sendmail` configuration files; previously, `sendmail` used Version 1 `sendmail` configuration files. You cannot use an earlier version `sendmail` configuration file with HP-UX 10.20 `sendmail`. There are three options available for migrating your configuration file:

- Make any modifications that you need to the `/etc/mail/sendmail.cf` file that is installed with HP-UX 10.20. This method is recommended where you have minimal site-specific changes to the `sendmail.cf` file.

(Note that if there is an existing Version 6 `/etc/mail/sendmail.cf` file, that file is not overwritten by the HP-UX 10.20 installation. If an existing `/etc/mail/sendmail.cf` file is not Version 6, this file is first copied to `/etc/mail/#sendmail.cf` and then the copy of the sample Version 6 file is written to `/etc/mail/sendmail.cf`. In either case, a copy of the sample Version 6 configuration file can be found in `/usr/newconfig/etc/mail/sendmail.cf`.)

- Create a new `sendmail.cf` file using the `m4` macros. For information on how to do this, refer to the `/usr/newconfig/etc/mail/cf/README` and `/usr/newconfig/etc/mail/cf/README.hpux10` files.
- Use the `convert_awk` utility to convert the old configuration file into the format required by HP-UX 10.20 `sendmail`. Note that while the converted file is usable by HP-UX 10.20 `sendmail`, it does not allow you to use the format and options available with the HP-UX 10.20 `sendmail.cf` file. Thus, you should only choose this migration option if your existing `sendmail` configuration file contains many site-specific rulesets that are not easily redefined in the HP-UX 10.20 `sendmail.cf` format.

Frozen files were hashed versions of the configuration file `sendmail.cf`. Frozen files were often problematic and are no longer a part of `sendmail`. Thus, both `/etc/freeze` and the `-bz` option have been removed. The standard `sendmail.cf` file is used instead.

HP no longer supports the `/etc/mail/rev-aliases` file that allowed a user to rewrite the “From:” line on outgoing mail based on a lookup to that `rev-aliases` database. Instead, the `userdb.db` file can be used.

Prior to HP-UX 10.20, `sendmail` tried to connect to a host directly if there were no MX records for the destination. To revert to this behavior, you must uncomment the option `TryNullMXList`.

There are no differences between the s700 and s800 versions of `sendmail`.

## Performance

The removal of `sendmail.fc` (the frozen version) should have only a nominal affect on `sendmail`'s performance. The database support improvements to `.db` files should help to offset this performance difference. Also, the improvement of timeout controls mentioned earlier should permit better tuning.

## sendmail for 10.0

The remote mode of `sendmail` was enhanced at 10.0. Delivery of out-bound mail messages and receipt of in-bound mail messages have been modified to support 8-bit data for electronic mail messages using E-SMTP and to handle non-“DUX” environments and the 10.0 file system layout. The Remote-mode feature was added for users on `sendmail` clients.

For more information on `sendmail`, refer to the `sendmail(1M)` manpage.

## Variable Length Subnet Masks

*For 10.0:*

Subnets of the same IP network number that have different sizes (that is, different masks) are called variable length subnets. The use of variable length subnetting techniques can help defer the onset of 32-bit IP address space exhaustion on the Internet by slowing down the need for allocating new IP network numbers. Packets within these subnets are routed to the best (longest or most specific) match.

The following variable length subnet features have been added at 10.01:

- Support for allowing a system to connect to variable length subnets that may have the same network address.

## 6-12 Networking

These subnets may be directly connected through the system's network interfaces or indirectly connected through a gateway. HP-UX systems running versions of the operating system prior to 10.01 assume one subnet mask for all network interfaces and routes that have the same network address.

- Support for Classless Inter-Domain Routing (CIDR), as specified in RFC 1519.

To delay deleting the 32-bit IP address space on the Internet, a new scheme called Classless Inter-Domain Routing (CIDR) has been standardized on to better utilize the class C address space. The use of subnet masks has been expanded: They may now cover part of the IP network portion of the IP address in addition to the IP host portion to create a "supernet."

Internet authorities responsible for assigning Internet addresses have started assigning large blocks of Class C addresses to companies that request them. The HP-UX variable length subnet masks feature lets these companies form company-wide "supernets" by using a single subnet mask for all systems in the network. Without it, these companies would have to break up their company networks into multiple subnets and use very complicated routing configurations to connect these subnets together.

- Support for OSPF Version 2, a standardized Internet Routing Protocol.

To support OSPF version 2, HP-UX systems will keep subnet masks in their routing table entries and adopt a longest match searching scheme.

For more detailed information, refer to the *Installing and Administering LAN/9000 Software* manual (98194-90057) and the Internet Routing manpages: `route(1M)`, `routing(7)`, `netstat(1)`, `ifconfig(1M)`, and `ppl.remotes(4)`.

## Domain Name Server (DNS) Resolver Changes

*For 10.0:*

If DNS is used on a 10.x host, there are a few small differences in behavior between 9.x and 10.x systems; these differences are especially noticeable if the domain keyword is used instead of the search keyword in `resolv.conf`, or if the local domain is generated from the system's hostname (when no `resolv.conf` is present).

If the domain keyword is used in the `resolv.conf` file, a multi-level search list is no longer generated by default, and partially qualified hostnames may not work with most networking commands as they did with 9.x. For example, on 9.x, `domain mkt.aone.com` creates the search list `mkt.aone.com aone.com`. On 10.x, `domain mkt.aone.com` creates the search list `mkt.aone.com`. To generate 9.x behavior on 10.x, you need to replace the domain keyword in the `resolv.conf` file with the search keyword (in this example, use `search mkt.aone.com` instead of `domain mkt.aone.com`).

If you are already using the search keyword instead of the domain keyword, the behavior will be the same as in 9.x. However, if you do convert from the domain keyword to a search keyword, read RFC 1535 (located in `/usr/share/doc`) to familiarize yourself with the security concerns.

The second change is that hostnames with a dot (.) in them are tried first without appending any of the resolver search list items. This happens whether or not the hostname was provided in a partially or fully qualified form. The default is to do this when at least one dot is present in the name; however, this value is adjustable with the `options` keyword and the `ndots` option. If you use this option and the number of dots does not reach the threshold, the search list items are appended. If they then fail, the name is tried without appending any domains. Note that if you put a dot at the end of hostname, the resolver does not use the search list and the name is tried without appending any domains to it.

Refer to the `resolver(4)` and `hostname(5)` manpages for more information on these changes.

If the DNS resolver API is used, a small change is necessary for programs using the resolver structure including the declaring of the global library structure `_res`. The resolver structure type (defined in `/usr/include/resolv.h`) has been renamed from `state` to `_res_state`. This change does not affect binary compatibility.

Refer to the `resolver(3)` manpage for more information.



## **mouted (DVMRP)**

*For 10.10:*

The `mouted` daemon, a new feature in HP-UX 10.10, is a routing daemon that forwards IP multicast datagrams, within an autonomous network, through routers that support IP multicast addressing. The `mouted` daemon implements DVMRP (the Distance-Vector Multicast Routing Protocol). DVMRP is an IGP (Interior Gateway Protocol) whose key purpose is to maintain the shortest return paths to the source of the multicast datagram.

The `mouted` daemon :

- intelligently handles the forwarding of multicast datagrams. It uses a “pruned” broadcast delivery tree so that messages are only forwarded to subnets that have identified themselves as having members of the destination multicast group.
- provides a mechanism called “tunneling” to handle multicasting among subnets separated by routers that do not support IP multicasting.
- handles multicast routing only. A separate process is required to handle unicast or broadcast routing.

For detailed information on `mouted`, refer to Chapter 8 “Configuring `mouted`”, of the *Installing and Administering Internet Services Manual* (B2355-90100) and to the `mouted(1M)` manpage.

## **Name Service Switch**

*For 10.10:*

The Name Service switch now allows the system administrator to configure the name service policy for additional network databases in addition to the host and IP address database. The new databases include: networks, protocols, services, rpc and netgroup. These new databases currently can use NIS or the local file as its data source.

The administrator can make changes to the switch policy by directly editing the configuration file. See the `switch(4)` manpage, which describes the configuration file syntax and the various settings available.

All code changes are contained within `libc`, thus any application that calls the routines to access services, protocols, networks, `netgroup`, and `rpc` databases can transparently gain this additional functionality. All applications built using shared libraries or the 10.10 archive libraries will indirectly use the new switch feature through these databases' access routines (for example, `getnetbyname()` and `getservbyname()`). The default configuration generates a behavior similar to that of previous HP-UX releases.

Any application that accesses NIS or the local database files directly, or are built on older archived libraries, need to be modified to make use of the new switch feature.

The `netgroup` database lookup routines `setnetgrent()`, `getnetgrent()`, `endnetgrnt()`, and `innetgr()` now support the Name Service Switch.

The Name Service Switch now determines where your system will look for the information that is traditionally stored in the following files:

```
/etc/hosts
/etc/protocols
/etc/services
/etc/networks
/etc/netgroup
/etc/rpc
```

In previous releases, the Name Service Switch determined only where your system looked for `/etc/hosts` information.

For all types of information except `/etc/hosts` information, you can configure your system to use NIS (one of the NFS Services), the local `/etc` file, or both, in any order. For `/etc/hosts` information, you can configure your system to use BIND (DNS), NIS, the `/etc/hosts` file, or any combination of the three, in any order.

If you use the default Name Service Switch configuration, your system will behave the way it did before the Name Service Switch was available, so you do not have to configure the Name Service Switch. Enter `man 4 switch` for more information.

*For 10.0:*

The 10.0 Name Service Switch feature allows you to configure the name service policy for hostname and address lookups. This policy is used by the algorithms

## **6-16 Networking**

in the `gethostbyname()` and `gethostbyaddr()` library calls. You can choose any combination of the supported name services: DNS, NIS, and the local hosts file. Also, you can set the conditions under which an alternative name service is to be used. The conditions that can be controlled on a per-name service basis include name services that are unavailable, that do not contain an entry for the host, or that are non-responsive.

You can make changes to the switch policy by directly editing the configuration file. The new `switch(4)` manpage describes the configuration file syntax and the various settings available. The name service switch is also described in the *Installing and Administering Internet Services* manual (B1031-90000).

## **Talk Service (talk/ntalkd)**

*For 10.10:*

Talk is a new Internet service for 10.10 that allows users to communicate directly over the network. Two users connected through the talk service can send and receive text, typed simultaneously, on terminals at both ends.

For more information, see the `talk(1)` and `talkd(1M)` manpages.

## **gated**

*For 10.10:*

The OSPF (Open Shortest Path First) protocol is one of the routing protocols handled by `gated`. In HP-UX 10.10, OSPF MIB (Management Information Base) support has been added to `gated`. The OSPF MIB objects are accessed through an SNMP (Simple Network Management Protocol) subagent called `ospfagt`.

---

<b>Note</b>	To all <code>gated</code> users: if <code>gated</code> is started automatically at system bootup, <code>ospfagt</code> is also started automatically.
-------------	---

---

The OSPF MIB can be accessed through the HP Network Management OpenView MIB browser. You can create OpenView applications using the information provided by the MIB browser. (HP does not provide any OpenView applications.)

In general, OSPF MIB provides useful status and statistical information related to the OSPF protocol. For more information on OSPF MIB objects, refer to RFC1253 (OSPF Version 2 Management Information Base).

The addition of OSPF MIB support does not create any compatibility issues with previous versions of `gated`, does not require any additional `gated` configuration to access this new feature, and has no effect on `gated` performance.

For more information on using the OSPF MIB feature, refer to the section titled “Accessing the OSPF MIB”, in Chapter 7, “Configuring `gated`”, of the *Installing and Administering Internet Services Manual* (B2355-90100).

*For 10.0:*

`gated` is a routing daemon that handles multiple routing protocols. It uses RIP, HELLO, and OSPF to collect information from within an autonomous system, and it uses BGP and EGP to advertise routes to another autonomous system. The `gated` daemon can be configured to perform all of the supported protocols or any combination of them. For more information on `gated`, refer to the `gated(1M)` manpage. Also, note that although BGP is provided in `gated`, it is not officially supported by Hewlett-Packard on releases 9.x, 10.0, or 10.01.

At HP-UX 10.0, support for the OSPF (Open Shortest Path First) protocol was added. OSPF uses a link state routing algorithm to determine routes. Each router on the network maintains an identical database describing the autonomous systems topology. Using the database information, each router constructs its own routing table of shortest paths from itself to each destination in the Autonomous System. OSPF overcomes the limitations of RIP and provides capabilities not found in RIP. With OSPF, a shorter convergence time can be achieved.

Variable length subnet mask support has been added to `gated` at 10.01. Networks can be partitioned into different sizes of subnets. If the networks within an autonomous system or domain continue to have subnets of the same size, no configuration changes are required. Otherwise, minor modifications may be needed. There are few mask statements in the `gated` configuration, and they are all optional. Two mask statements are under the OSPF protocol specification, and both are used to aggregate subnet routes: The mask under the `area networks` statements is used only by the area border routers, and the mask under the `backbone networks` is used by the backbone routers. Note that

you should avoid mixing networks that are partitioned into variable length subnets (with the support of the variable subnet masks) with networks that do not have this support in an autonomous system or domain. Nodes without this support do not contain the subnet mask information in the kernel routing table and therefore may fail to route packets to their proper destination. For more detailed information, refer to the *Installing and Administering Internet Services* manual (B1031-90000) and the `gated.conf(4)` manpage.

## Boot Relay Agent

*For 10.0:*

The `bootp` relay agent, a new feature at 10.0, is provided as a part of the `bootp` daemon to enable booting across gateways. When a client resides on a different subnet/LAN from the server, a `bootp` relay agent is needed to run on a host on the same subnet as the client. When the client sends out a `bootp` broadcast request, the relay agent relays the `bootp` request to the `bootp` server and also relays the reply back to the client.

The `bootptab` file is used as the default configuration file for configuring the `bootp` relay agent (that is, the server(s) to relay the `bootp` request to). Changes to the relay agent configuration need to be made manually in the `bootptab` configuration file. Additional information on the `bootp` relay agent function and configuration is in the `bootpd(1M)` manpage.

## Trivial File Transfer Protocol Server

*For 10.0:*

The following two enhancements were added to the TFTP server at 10.0:

- An additional way of restricting access to files and directories on the server system. Previously, `tftpd` allowed access only to files located in the home directory of user “tftp.” To activate the service, you had to create an account for user “tftp” and copy the files that could be the object of TFTP transfers to that location. In 10.x, you can instead specify a list of directories and/or files as arguments to the `tftpd` command in the `/etc/inetd.conf` file. The two methods can be combined, making it possible to use the new feature without disturbing existing TFTP configurations.

- Temporary bypassing of `inetd` after the initial request. The TFTP server is started by `inetd` as before. However, after handling the initial request, `tftpd` stays up for awhile, accepting new requests directly. This new behavior helps the process of booting many systems or X terminals from one server.

For more information, refer to the `tftpd(1M)` manpage.

## Remote Distribution Service (`rdist`)

*For 10.0:*

The `rdist` service was added at 10.0. `rdist` is designed to distribute files from a central server. It simplifies the task of maintaining identical copies of files on multiple systems. For example, you can modify files on the server, and then use `rdist` to propagate the files to the other systems in the network.

For more information, see the `rdist(1)` manpage and the the section on `rdist` in the *Installing and Administering Internet Services* manual (B1031-90000).

## Network Time Protocol

*For 10.0:*

Network Time Protocol (NTP), which is a time synchronization service for the Internet architecture, was added at 10.0. NTP provides a mechanism to synchronize time and coordinate time distribution in a large, distributed internet. NTP operates in various modes to improve efficiency. A configuration file is used to configure NTP execution parameters such as communication mode with other NTP servers. The default configuration file is `/etc/ntp.conf`.

Network Time Protocol Query (`ntpq`), a tool to query the status of NTP servers, has also been added. It can be run in command-line mode as well as in interactive mode.

Network Time Protocol Date (`ntpdate`) is a tool to set the local host's date and time by polling the Network Time Protocol Servers.

Descriptions of the NTP functions and configurations, `ntpq`, and `ntpdate` are in the following manpages: `xntpd(1M)`, `ntpq(1M)`, `ntpdate(1M)`.

## File Transfer Protocol Service

*For 10.0:*

Two new options to the File Transfer Protocol service (`ftp/ftpd`) were added at 10.0:

- The `ftp` server and client support the `-B` option, which allows you to set the buffer size for the data socket to any value from 1 KB to 64 KB (the default is 56 KB).
- The `ftp` server supports the `-v` (verbose) option, which turns on logging of information that was previously logged only for anonymous `ftp`.

For more information, refer to the `ftp(1)` and `ftpd(1M)` manpages.

## nslookup

*For 10.0:*

`nslookup`, the diagnostic tool for name service resolution, was updated at 10.0 to recognize the name service switch (see the previous information on “Name Service Switch” in this section) and to provide some limited switch-related diagnostics. These changes include the addition of three new interactive-mode commands:

- The `policy` command prints out the current switch policy of the local host.
- The `server` command is now available when the `nslookup` tool uses either NIS or `/etc/hosts` as the first name service in the policy. In these cases, the command overrides the switch policy to directly query the specified name server.
- The `reset` command has been added, and you can use it to return the tool to using the switch policy.

Two new options have also been added for tracing actions affected by the switch:

- The **swtrace** option prints out the actions of switching between name services according to the specified policy.
- The **swdebug** option (available only on the command line) prints out the parsing of the name service switch configuration file.

For more information on **nslookup**, refer to the **nslookup(1)** manpage.

## **mailx**

*For 10.0:*

Two new headers in every **mailx** message to support 4-byte EUC and MIME were added at 10.0:

```
Content-Type: text/plain; charset=X
Content-Transfer-Encoding: Y
```

For more information on **mailx**, refer to the **mailx(1)** manpage.

## **telnet**

*For 10.0:*

10.x **telnet** supports SD (see “Software Distributor” in Chapter 4), the 10.0 file system layout conventions (see “The HP-UX 10.0 File System Layout” in Chapter 4), and port identification device files in a special directory (**/dev/telnet**). HP-UX 10.x includes a script (**/usr/contrib/bin/ddfa\_dev\_mig**) that will help you move device files to this special directory.

In addition, **telnet** supports the following new options at 10.01:

- Option negotiation—Changes have been made to process the new options.
- Terminal speed option—The remote application can specify the common terminal speed.
- Flow control option—The kernel has been modified to detect the flow control on/off request and modify the **pty** setting.

## **6-22 Networking**



- User id option—The `userid` is passed on to the remote node, which verifies the identity of the user to check if the user has an authorized login in its system, and then opens a new `telnet` session.

For more information on `telnet`, refer to the `telnet(1)` and `telnetd(1M)` manpages.

## **whois**

*For 10.0:*

The `whois` command is new for 10.01. It is somewhat similar to a phone book lookup utility, and it is of interest primarily to military and government users. `whois` queries a database that is kept at the Network Information Center (NIC) and returns information about persons and organizations that are listed there. You must be able to make a direct socket connection to the NIC in order to get any response. The database at the NIC is often heavily loaded, so response can be slow. Also, vague or ambiguous requests (for example, “Smith”) can take a long time to process and produce hundreds of pages of output. For more information, refer to the `whois(1)` manpage.

## **DDFA**

*For 10.0:*

10.x DDFA supports SDU and the 10.0 file system layout conventions, as well as having pseudonym device files in a special directory (`/dev/telnet`). The `/usr/contrib/bin/ddfa_dev_mig` script facilitates moving device files to this special directory. DDFA also has more readable error messages and a way to filter the severity level of messages to be displayed. In addition, DDFA allows ARPA node names in place of IP addresses.

DDFA now supports outgoing connections to devices on non-HP terminal servers, provided they use one of the supported options for defining the output device in the `dp` configuration file. These options include:

- For DTCs only, using the IP address or node name, and board/port number.
- For most terminal servers, including DTCs, specifying the server IP address or node name, and its unique port TCP address.

- For the DTC and servers, specifying just the server IP address or node name so that connections go to the default TCP address of 23.

For additional information, refer to the `dpp(1M)` and `ocd(1M)` manpages.

## ARPA Transport

*For 10.0:*

NetIPC, a Hewlett-Packard proprietary networking Application Programming Interface (API), is not supported on HP-UX 10.x. All of the NetIPC system calls, the `sockreg` daemon, `rlb` daemon, `rlb` diagnostic, `nodename` command, `proxy` command, and PROBE proxy services are not supported. HP-UX 10.x has no concept of the NS nodename space. As a result, HP-UX 10.x will not use PROBE to resolve name-to-address mapping and will not respond to PROBE name requests from other machines (HP 3000s, HP 1000s, PCs, and pre-10.0 HP 9000s). However, HP-UX 10.x will respond to PROBE Virtual Network Address (VNA) requests.

All applications and services that use NetIPC need to be migrated to BSD Sockets (unless you do not want to update them to HP-UX 10.x). HP integration sockets, `vt3k`, X.400, and AllBase/Net are still supported, but `dscopy` is not. For more information on NetIPC, refer to the *NetIPC to BSD Sockets Migration Guide* (98194-90045). For additional ARPA transport information, refer to the *Installing and Administering LAN/9000 Software* manual (98194-90057).

New features added to TCP/IP at 10.0 include the following:

- High-performance TCP extensions (RFC 1323) for high-speed links (FDDI and Fiber Channel).
- Path MTU (PMTU, RFC 1191).
- Compliance with *Requirements for Internet Hosts*, RFC 1122.
- Support for CSLIP as well as SLIP, RFC 1144.
- Support for multiple IP addresses per interface and local and remote switch-over between interface cards (supplied as part of the MC/ServiceGuard product).
- Support for the Router Discovery Protocol (RDP, RFC 1256).

## 6-24 Networking

- Support for IP Multicasting, including level 2 IGMP (RFC 1112).
- XPG4-version of XTI over TCP/UDP/IP.
- A new network node-management tool called `net tune`. It is used to modify certain kernel parameters that control networking defaults such as keepalive timers. `net tune` is in `/usr/contrib/bin`.
- A public-domain version of `traceroute`, located in `/usr/contrib/bin`.
- Additional protocol-level logging through the NetTL subsystem.

---

## LAN/9000

### 48 LAN Card Support

*For 10.10:*

The Series T500 (part of the S800 family) will support up to 48 LAN cards as follows:

- 32 Ethernet/802.3
- 8 FDDI
- 8 Token Ring/802.5

No changes are needed for the ARPA Transport code. However it is highly recommended that the administrator run `net tune` and adjust the maximum IP interrupt queue parameter `ip_intrqmax` to optimize performance. The default value is 50.

This feature now expands the maximum number of LAN cards that can be installed on a T500 system. One utilization of this feature will be for High Availability, enabling half the LANs (for example, 16 Ethernet and 4 FDDI interfaces) while keeping the remainder as backup in case of failure.

This feature requires additional NIO expander bays for additional LAN cards. Each expander bay can handle 14 half-height (Ethernet and Token Ring) or 7 full-height (FDDI) cards.

## LANCONFIG

At 10.0, a new feature was added to the `lanconfig` command to allow users of TCP/IP over the HP-UX Token Ring/9000 products to selectively disable or enable the Source Routing feature for 802.5. This produces additional command-line options. For more information, refer to the `lanconfig(1M)` manpage.

## LANSCAN/LANADMIN/LINKLOOP

At HP-UX 10.0, there were a number of changes to the LAN commands:

- The display format of `lanscan` has changed, and there are new options available.
- The `lanadmin` command replaces the `landiag` command. `lanadmin` has additional features that support setting the set link speed, station address, and MTU size. The `landiag` command exists as a softlink to `lanadmin`.
- The `linkloop` command requires new parameters to specify the selected interface.

For more information, refer to the *Installing and Administering LAN/9000 Software* manual (98194-90057) and the `lanscan(1M)`, `lanadmin(1M)`, and `linkloop(1M)` manpages.

## NETTUNE

At HP-UX 10.0, there was a new command called `net tune` that provides an interface for configuration and tuning of various network parameters. `net tune` supports a consistent, reliable, and secure method for you to use to tune network parameters based on your performance and configuration needs. `net tune` allows you to view (`get`) or change (`set`) the value of many networking parameters, such as the TCP keep-alive timeout value(s), default socket buffer sizes, or the enabling/disabling of IP forwarding. All users may view the values of supported configurable network parameters, but only those with super-user capability can update a parameter. `net tune` is command-line driven and has an online help facility that briefly describes the available configurable values. `net tune` is in `/usr/contrib/bin`, and is not supported by Hewlett-Packard.

For more information on `net tune`, refer to the `net tune(1)` manpage in `/usr/contrib/man/man1`.

## LAN Product Keywords

The keywords in the [78]00 system configuration file for the LAN products changed at HP-UX 10.0. Also, dependency features for the LAN products have changed. The LAN products require DLPI and STREAMS modules to be present in the kernel.

- The `lan` and `lan01` keywords are no longer supported. You need to use `lan2` for Series 700 system configuration files, and `lan0`, `lan1`, `lan2`, and/or `lan3` for Series 800s.
- The DLPI keyword along with STREAMS keywords (`hpstreams`, `timod`, `tirdwr`, `clone`, `strlog`, `sad`, and `echo`) must be included in system configuration files.

If these changes are not made, kernel build failures, either in `conf.c` failures or load failures for `vmunix`, may occur.

For more information, refer to the *Installing and Administering LAN/9000 Software* manual (98194-90057) and the `ifconfig(1M)` and `lanscan(1M)` manpages.

---

## LAN Cable Disconnect

The `lan2` (Core/Lasi/EISA 802.3), `lan3` (HP-PB 802.3), `lan6` (HP-PB FDDI), and `token2` (HP-PB 802.5) products have new LAN cable disconnect functionality at 10.0, which enables the card and driver to detect a cable-disconnect problem if NetTL logging is enabled for the various products involved. When a possible cable disconnect occurs, the driver enters a monitoring state. During this state, the driver discards any further packets received from any of the networking layers above—LLA, DLPI, or transport. The driver also sets `Net_Interface` to `DOWN` if it was `UP` at the time the disconnect was detected. When the card and driver detect that the writes onto the media are working (the cable appears to be functioning), the driver changes

state, resets the `Net_Interface` to the state before the disconnection, and begins to process reads and writes.

For additional information, refer to the *Installing and Administering LAN/9000 Software* manual (98194-90057) and the `lanscan(1M)` and `netstat(1)` manpages.

---

## NETTRACELOG

At 10.0, the NETTRACELOG fileset was divided into two filesets: NETTL-MIN and NETTL-RUN. For 10.01, a third fileset, NETTL-ADM, has been added.

- NETTL-MIN contains the `nettlgen.conf` configuration database, the `nettlconf` script, and the `libntl.sl` shared library. The NETTL-MIN fileset is necessary to allow products to be configured in the network tracing and logging configuration database and for user-space clients to make calls to the tracing and logging facility.
- NETTL-RUN contains the runtime portion of the network tracing and logging facility: the `nettl` and `netfmt` commands, libraries, header files, daemons, device files, and start-up scripts. If the NETTL-RUN fileset is removed, no tracing and logging functionality is available, but clients of network tracing and logging continue to function normally.
- NETTL-ADM contains a new NetTL interactive user interface called `nettladm`, which is provided in addition to the `nettl` and `netfmt` commands. The `nettladm` user interface runs graphically using the X11/Motif environment if the `DISPLAY` environment variable is set to a host that is running an X server. The interface can also run in a text terminal environment.

The `nettladm` command starts a menu-driven program that makes it easy for users (even those with limited specialized knowledge of HP-UX) to perform tracing and logging tasks. It is a self-guided tool, and users can get context-sensitive help at any point by pressing the `f1` function key. `nettladm` allows modification of network logging and can be used to control the information that is displayed on the system console. It can be used to capture network traces and facilitate the analysis of the collected data.

Although root access is required to make changes to logging or tracing, non-root users are allowed to make status, monitoring, and data formatting operations.

If you do not want to use `nettladm`, you do not need to install the NETTL-ADM fileset. Also, you can remove the fileset at any time without impacting the operation of your system.

In addition, there have been changes to some of the NetTL commands at 10.0. For more information, refer to the `nettl(1M)`, `nettlconf(1M)`, and `netfmt(1M)` manpages.

---

## Networking `_r` Routines

*For 10.20:*

The networking `_r` routines provide thread-safe mechanisms for accessing network database information either via the `/etc` directory or NIS. The non `_r` routines will be modified to be thread safe.

Usage of the networking `_r` routines defined below will no longer be supported in a future HP-UX release. In that future release, the `_r` routines will exist, but there might be semantic changes to the interfaces. That is, there might no longer be a per thread view of the `/etc` database and all fields in the `struct API_data` will be unsupported.

### Compatibility

Binary compatibility is affected if you are accessing any part of the `struct API_data` after the call to the network `_r` routines. In a future HP-UX release, these fields will no longer contain valid information.

Examples of `API_data` are:

- `hostent_data`
- `netent_data`
- `servent_data`
- `protoent_data`
- `rpcent_data`

See `/usr/include/netdb.h` for the fields of these structures.

## Changes Affecting Routines

The following routines will be moved to a new library in a future release. Also, the routines may be obsoleted in a future HP-UX release.

<code>endhostent_r</code>	<code>endnetent_r</code>	<code>endnetgrent_r</code>
<code>endprotoent_r</code>	<code>endrpcent_r</code>	<code>endservent_r</code>
<code>gethostbyaddr_r</code>	<code>gethosbyname_r</code>	<code>gethostent_r</code>
<code>getnetbyaddr_r</code>	<code>getnetbyname_r</code>	<code>getnetent_r</code>
<code>getnetgrent_r</code>	<code>getprotobyname_r</code>	<code>getprotobyport_r</code>
<code>getprotoent_r</code>	<code>getrpcbyname_r</code>	<code>getrpcbynumber_r</code>
<code>getprcent_r</code>	<code>getservbyname_r</code>	<code>getservbyport_r</code>
<code>getservent_r</code>	<code>sethostent_r</code>	<code>setnetent_r</code>
<code>setnetgrent_r</code>	<code>setprotoent_r</code>	<code>setrpcent_r</code>
<code>setservent_r</code>		

---

## NFS

For HP-UX 10.x, NFS is configured through the files in `/etc/rc.config.d` and not through modifying `rc` scripts directly. A new parameter (`-m`) was added to the `nfsstat` command at 10.0, and it is described in the `nfsstat(1M)` manpage. Refer to the *Installing and Administering NFS Services* manual (B1031-90000) for more information.

HP-UX 10.x provides the following features of NFS 4.2:

- The `fuser(1M)` command accepts NFS file systems.
- Support for the Loopback File System (LOFS) has been added.

LOFS allows mounting an existing directory onto another directory. The effect is similar to that of a symbolic link, except that the usual mount options are available, and there are no issues with the meaning of `..`. When an application does a `stat(2)` on a LOFS file, the values returned are the same as for the file in its original location, except that the `st_fstype` field indicates LOFS. In particular, `st_dev` and `st_ino` are the same, so that the conventional test for whether two pathnames name the same file (device and inumber match) continues to work.



Some applications that walk file trees use `stat`'s `st_dev` to detect the crossing of mount points. This will not work with LOFS because an LOFS file has the `st_dev` of its original location. Code that needs to identify a particular file system (in the sense of a particular mount) should use `f_fsid` from `statfs(2)` or `statvfs(2)`.

## NFS Commands

The following is a summary of 10.0 NFS changes:

- The RPC header files (`/usr/include/rpc`) were changed to support ANSI C and C++ compilations.
- Automounter is now required to have a `netgroup` entry that matches the `netgroup` entry on the NFS server to mount the exported file system.
- `stdhosts` prints the following warning message to `stderr` if the first entry on a line in `/etc/hosts` does not have the correct syntax for an IP address:

```
stdhosts: Warning: malformed line ignored:  
< the line in question >
```

Correct the “line in question” so that it begins with a valid syntax IP address or a comment (`#`) character.

- `rpc.yppasswdd` now `cds` into `/var/yp` and issues a `make` command instead of a `yppmake` command.
- The `nis.client` control script no longer unsets the domain name if a `ypserv` process that supports the configured domain cannot be found.

---

## NIS

*For 10.20:*

NIS APIs, specifically `yp_bind()`, and commands will now time out if a server is not available.

The `nis.client` startup script has also been modified so that it no longer hangs during a boot if an NIS server is unavailable. If the script cannot bind to a server, NIS will be disabled.

A failure to bind to a server during the boot sequence will cause the `nis.client` start-up script to disable NIS and then continue the boot process.

*For 10.10:*

NIS configuration was modified at 10.0 as follows:

- It starts the `ypxfrd` process on all NIS masters and slaves.
- It starts the `rpc.yupdated` process on all masters.
- It starts the `keyserv` process on all machines.
- You can customize what action a client should take if an NIS server is not found. You can force the client to wait for a server, or allow the client to boot without NIS.
- HP-UX 10.x `RPCGEN` is equivalent to `RPCGEN` in Sun Microsystem's `RPC 4.2`. (`RPCGEN` is the protocol compiler for `RPC` client/server applications.) Several new command-line options (`-D`, `-I`, `-K`, `-L`, `-T`, and `-t`) have been added to `RPCGEN`. For more information, refer to the `rpcgen(1)` manpage.
- The `NLS` feature is not supported on 10.x.
- The `clnt_create_vers()` function from `ONC 4.2` has been added. This expands the library of `RPC` calls for `RPC` applications. For more information, refer to the `rpc_clnt_create(3c)` manpage.
- `Secure RPC` has been added. Refer to the `secure_rpc(3c)` manpage for more information on `secure RPC` and descriptions of the library routines for `secure remote procedure calls`.

The following NIS commands were new or changed at 10.0. Refer to the manpages shown for more information:

**Table 6-1. NIS Commands**

Command	Status	Manpage
ypbind	Revised	ypserv(1M)
ypserv	Revised	ypserv(1M)
ypxfrd	New	ypserv(1M)
ypxfr	Revised	ypxfr(1M)
yppush	Revised	yppush(1M)
rpc.yupdated	New	ypupdated(1M)
keyserv	New	keyserv(1M)
keyenvoy	New	keyenvoy(1M)
keylogin	New	keylogin(1)
keylogout	New	keylogout(1)
newkey	New	newkey(1M)
chkey	New	chkey(1M)
ypinit	Revised	ypinit(1M)
ypmake	Revised	ypmake(1M)
makedbm	Revised	makedbm(1M)
yp_update	New	yp_update(3c)

---

## NS/9000 and vt3k/9000

The `dscopy` (`nft`) service of NS/9000 is not supported on HP-UX 10.x. This includes the `dscopy` command and the `nftserver`, `nftdaemon`, and `dscopy` processes. The `ver_ns` verification script is also not supported. Applications and services currently using `dscopy` will need to use the Internet Services File Transfer protocol (`ftp`) for 10.x. Other NS systems in the network may need to be updated to maintain `vt3k` communication with HP-UX systems running 10.x. HP 3000 names need to be configured in the `/etc/host` file or on the Domain Name Server.

The `vt3k` service is supported on HP-UX 10.x, but it uses the BSD Sockets interface. With the transition to BSD Sockets, `vt3k` uses ARPA Domain Name resolution rather than NS nodename resolution. The `-t` option of `vt3k` now enables MPE/iX-style typeahead. You may need to modify scripts that use this option and that run HP 3000 applications not designed for typeahead (for example, applications that use block mode or function keys) so that they invoke `vt3k` without the `-t` option. For more information, including HP 3000 software requirements, refer to the *VT3K/9000 Quick Reference Manual* (J2399-90000).

---

## ONC/NFS Large UID Support

*For 10.20:*

The ONC/NFS software (RPC/XDR, YP, NFS) has been modified to support the large UID changes in 10.20. Specifically, API interfaces and header files have been modified to support the `uid_t` and `gid_t` structures.

ANSI C or C++ compilers may fail due to changes in header files. There is no effect on binary compatibility.

A source compatibility issue exists where a function prototype has been changed to support the `uid_t` and `gid_t` struct instead of using either `long` or `int`. A compilation error will occur when the parameter type no longer matches the prototype definition.

Example of a typical failure:

```
func_one(gid_t *) /*prototype*/  
  
.  
.  
  
    long gid;          /* This now needs to be uid_t gid */  
    func_one(&gid);  
  
.  
.
```

---

## PMTU

The HP-UX 10.x networking kernel includes Path Maximum Transmission Unit (PMTU) Discovery for improving networking performance. PMTU is a technique for determining the maximum transmission unit of an arbitrary Internet path dynamically without fragmentation. Both the host and router implementations for PMTU are included in HP-UX 10.x. By default, the networking kernel enables the PMTU algorithm for all TCP connections.

For more information on PMTU, refer to the *BSD Sockets Interface Programmer's Guide* (98194-90052) and the `route(1M)`, `netstat(1)`, `getsockopt(2)`, and `ping(1M)` manpages.

---

## SLIP and CSLIP

### SLIP

On 10.0 and 10.01, there is a limitation with the SLIP product. In a dial-in with login environment, the command line `"/usr/bin/pp1\s -i\host running` (as configured in the `pp1.remotes` file) may fail to be executed by the dial-in host, if the login shell is `sh` or `ksh`. You need to use one of the following workarounds; the first one is *highly recommended*.

- *On the dial-in system:* Use `cs` as the login shell for the slip login. SLIP itself does not care which shell it uses.
- Put in an expected prompt between the double quotes. This must be done on *all* clients. In network with a large amount of users, this could be prohibitive.
- Remove `ttytype` from the dial-in host's `/etc/profile`. This workaround will cause problems with non-HP term types.

## **CSLIP**

HP-UX 10.x supports Compressed Header Serial Line Internet Protocol (CSLIP). CSLIP provides better performance than SLIP, especially for applications that send many small packets (such as `telnet`, SharedX, and X-Windows). However, if you use CSLIP, be sure that the systems you wish to connect to also use CSLIP; if they are using SLIP instead of CSLIP, you must use SLIP to communicate with them.

HP-UX 10.x supports CSLIP or SLIP dial-in connections over the DTC to a LAN-based host, provided that the DTC is configured to use binary mode via the OpenView DTC Manager, and the nailed DTC port address is configured using the DDFA utility. Speeds of up to 9600 baud on HP 9000 Series 800 hosts (only) are supported.

For more information on CSLIP and SLIP, including using CSLIP and SLIP for host-to-host communication over an asynchronous telephone line, refer to the *Using Serial Line IP Protocols* manual (98194-90051).

---

## Sockets

*For 10.10:*

HP-UX supports two sets of sockets: HP-UX Sockets (as of Release 10.01) and X/Open Sockets.

By default, your application will use HP-UX Sockets. To get X/Open Sockets, the macro `_XOPEN_SOURCE` must be defined and the macro `_XOPEN_SOURCE_EXTENDED` must be defined to 1.

The `_XOPEN_SOURCE` macro can be automatically defined by the compilation environment, but to ensure portability, the application should define the macro either on the compilation command line or at the beginning of each source module prior to the inclusion of any headers. The `_XOPEN_SOURCE_EXTENDED` macro will not be automatically defined by the compilation environment.

`make` scripts will also have to include “`-l xnet`” as an argument to the `c89` or `cc` utilities. The argument will link your applications to a new shared library `libxnet.sl` that contains both X/Open Sockets and XTI-conformant functions.

For a detailed description of X/Open Sockets and XTI, refer to *X/Open CAE Specification, Networking Services, Issue 4* or *Going Solo* published by Prentice Hall. *Going Solo* also includes a CD ROM that contains the above specification in addition to other UNIX95 manpages.

## Socket Interfaces

This section highlights the differences between HP-UX Sockets and X/Open Sockets.

### **accept**

*X/Open:*

```
int accept(int socket, struct sockaddr *address, size_t *address_len);
```

*HP-UX:*

```
int accept(int socket, void *address, int *address_len);
```

## **bind**

*X/Open:*

```
int bind(int socket, const struct sockaddr *address, size_t address_len);
```

*HP-UX:*

```
int bind(int socket, const void *address, int address_len);
```

## **connect**

*X/Open:*

```
int connect(int socket, const struct sockaddr *address, size_t address_len);
```

*HP-UX:*

```
int connect(int socket, const void *address, int address_len);
```

## **getpeername**

*X/Open:*

```
int getpeername(int socket, struct sockaddr *address, size_t *address_len);
```

*HP-UX:*

```
int getpeername(int socket, void *address, int *address_len);
```

## **getsockname**

*X/Open:*

```
int getsockname(int socket, struct sockaddr *address, size_t *address_len);
```

*HP-UX:*

```
int getsockname(int socket, void *address, int *address_len);
```

## **getsockopt**

*X/Open:*

```
int getsockopt(int socket, int level, int option_name,  
               void *option_value, size_t *option_len);
```

*HP-UX:*

```
int getsockopt(int socket, int level, int option_name,  
               void *option_value, int *option_len);
```



Two new options, `SO_ACCEPTCONN` and `SO_TYPE`, are added to HP-UX Sockets and X/Open. `SO_ACCEPTCONN` reports whether socket listening is enabled and returns an `int option_value`. The value will be 1 if the socket listening is enabled; otherwise it will be 0. `SO_TYPE` reports the socket type of `SOCK_DGRAM` or `SOCK_STREAM`.

### **recv**

*X/Open:*

```
ssize_t recv(int socket, void *buffer, size_t length, int flags);
```

`recv` supports a new flag, `MSG_WAITALL`, which requests that the function block wait until the full amount of data requested can be returned. The function may return a smaller amount of data if a signal is caught, the connection is terminated, or an error is pending for the socket.

*HP-UX:*

```
int recv(int socket, void *buffer, int length, int flags);
```

### **recvfrom**

*X/Open:*

```
ssize_t recvfrom(int socket, void *buffer, size_t length, size_t flags,  
                struct sockaddr *address, size_t *address_len);
```

*HP-UX:*

```
int recvfrom(int socket, void *buffer, int length, int flag,  
            void *address, int *address_len);
```

`recvfrom()` supports a new flag, `MSG_WAITALL`, as described in the section above on `recv()`.

### **recvmsg**

*X/Open:*

```
ssize_t recvmsg (int socket, struct msghdr *msg, int flags);
```

*HP-UX:*

```
int recvmsg (int socket, struct msghdr msg[], int flags);
```

X/Open Sockets `msghdr` has the following form :

```

struct msghdr {
    void      *msg_name;           /* optional address */
    size_t    msg_namelen;        /* size of address */
    struct    iovec *msg_iov;      /* scatter array for data */
    int       msg_iovlen;         /* # of elements in msg_iov */
    void      *msg_control;        /* ancillary data, see below */
    size_t    msg_controllen;     /* ancillary data buffer len */
    int       msg_flags;          /* flags on received message */
}

```

`msg_control` specifies a buffer to receive any ancillary data sent along with the message. Ancillary data consists of a sequence of pairs, each consisting of a `cmsghdr` structure followed by a data array. The data array contains the ancillary data message and the `cmsghdr` structure contains descriptive information that allows an application to correctly parse the data.

`cmsghdr` has the following structure:

```

struct cmsghdr {
    size_t    cmsg_len;           /* data byte count, including hdr*/
    int       cmsg_level;         /* originating protocol */
    int       cmsg_type;          /* protocol-specific type */
}

```

The supported value for `cmsg_level` is `SOL_SOCKET` and the supported value for `cmsg_type` is `SCM_RIGHTS`. Together, they indicate that the data array contains the access rights to be received. Access rights are supported only for `AF_UNIX`. Access rights are limited to file descriptors of size `int`. If ancillary data is not being transferred, set the `msg_control` field to `NULL` and set the `msg_controllen` field to 0.

The flags parameter accepts a new value, `MSG_WAITALL`, which requests that the function block wait until the full amount of data requested can be returned. The function may return a smaller amount of data if a signal is caught, the connection is terminated, or an error is pending for the socket.

On successful completion of `recvmsg()`, the `msg_flags` member of the message header is the bitwise-inclusive OR of all of the following flags that indicate conditions detected for the received message.

Flag	Description of Condition
<code>MSG_EOR</code>	End of record was received(if supported by the protocol).
<code>MSG_OOB</code>	Out-of-band data was received.

MSG\_TRUNC            Normal data was truncated.

MSG\_CTRUNC          Control data was truncated.

## **send**

*X/Open:*

```
ssize_t send (int socket, const void *buffer, size_t length, int flags);
```

*HP-UX:*

```
int send (int socket, void *buffer, int length, int flags);
```

## **sendmsg**

*X/Open:*

```
ssize_t sendmsg (int socket, const struct msghdr *msg, int flags);
```

*HP-UX:*

```
int sendmsg (int socket, const struct msghdr msg[], int flags);
```

*X/Open* Sockets uses a different `msghdr` definition. Refer to the section on `recvmsg()` for more details.

## **sendto**

*X/Open:*

```
ssize_t sendto (int socket, const void *msg, size_t length,  
               int flags, const struct sockaddr *dest_addr, size_t addr_len);
```

*HP-UX:*

```
int sendto (int socket, const void *msg, int length,  
            int flag, const void *dest_addr, int addr_len);
```

## **setsockopt**

*X/Open:*

```
int setsockopt (int socket, int level, int option_name,  
               const void *option_value, size_t option_len);
```

*HP-UX:*

```
int setsockopt (int socket, int level, int option_name,  
               const void *option_value, int option_len);
```

SO\_LINGER in X/Open defines `l_linger` as linger time in seconds, while SO\_LINGER in HP-UX defines `l_linger` as a qualifying flag indicating that `close()` should not return until all buffered data is sent.

## shutdown

The following will be available through `socket.h`:

- SHUT\_RD(0)
- SHUT\_WR(1)
- SHUT\_RDWR(2)

X/Open will generate [ENOTCONN] if the socket is not connected. HP-UX Sockets will generate [EINVAL]. Note that [EINVAL] in X/Open indicates how the argument is invalid.

## Sockets Header Differences

### <fcntl.h>

Two defines, `F_GETOWN` and `F_SETOWN`, will be visible to X/Open users.

### <sys/socket.h>

- For X/Open Sockets, the `sa_family` field in `sockaddr` will have an unsigned integral type, `sa_family_t`, which is equivalent to unsigned `short`, as in 10.01.
- The `msg_hdr` type in X/Open Sockets will be significantly different. Refer to the section on `recvmsg()` for more details.
- The following macro has been defined to aid traversing the ancillary data structures as defined in X/Open Sockets:

```
/* given pointer to msg_hdr, return pointer to the first cmsghdr. */
MSG_FIRSTHDR(mhdr)

/* given pointer to struct cmsghdr, return pointer to next cmsghdr */
MSG_NXTHDR(mhdr, cmsg)

/* given pointer to struct cmsghdr, return pointer to data */
```

`MSG_DATA(msg)`

- For X/Open Sockets, there is a new field, `msg_flags`, which indicates special conditions upon successful completion of the function involved, namely `recvmsg`. The following conditions are defined:

Flag	Description
<code>MSG_CTRUNC</code>	Control data truncated.
<code>MSG_EOR</code>	Terminates a record(if supported by protocol).
<code>MSG_OOB</code>	Out-of-band data was received.
<code>MSG_TRUNC</code>	Normal data was truncated.

- `l_linger` has different semantics in X/Open. It will be defined as the linger time in seconds.
- The following defines will be added for the `how` argument in the shutdown function: `SHUT_RD` (0), `SHUT_WR` (1), and `SHUT_RDWR` (2).
- Two new socket options, `SO_ACCEPTCONN` and `SO_TYPE`, are supported. They can be used in `getsockopt()`.

<**sys/stat.h**>

`S_IFSOCK` and `S_ISSOCK` will be visible to X/Open users.

## IP Address Resolution Interfaces

### **gethostent, gethostbyaddr, gethostbyname, sethostent, and endhostent**

*X/Open:*

```
struct hostent *gethostbyaddr(const void *addr, size_t len, int type);
```

*HP-UX:*

```
struct hostent *gethostbyaddr(const char *addr, int len, int type);
```

### **getnetent, getnetbyaddr, getnetbyname, setnetent, and endnetent**

*X/Open:*

```
struct netent *getnetbyaddr(in_addr_t net, int type);
```

*HP-UX:*

```
struct netent *getnetbyaddr(int net, int type);
```

### **htonl, htons, ntohl, and ntohs**

*X/Open:*

```
#include <arpa/inet.h>

in_addr_t htonl (in_addr_t hostlong);
in_port_t htons (in_port_t hostshort);
in_addr_t ntohl (in_addr_t netlong);
in_port_t ntohs (in_port_t netshort);
```

*HP-UX:*

```
X/Open #include <netinet/in.h>

unsigned long htonl (unsigned long hostlong);
unsigned short htons (unsigned short hostshort);
unsigned long ntohl (unsigned long netlong);
unsigned short ntohs (unsigned short netshort);
```

### **inet\_addr, inet\_network, inet\_makeaddr, inet\_lnaof, inet\_netof, and inet\_ntoa**

The network number, `net`, in the `in_addr` structure and the `inet_makeaddr()` function is defined as `in_addr_t` in X/Open. For HP-UX, it is defined as an `int` type. `in_addr_t` is defined as an unsigned 32-bit int.

## **IP Address Resolution Headers**

**<arpa/inet.h>**

`<arpa/inet.h>` will include `<netinet/in.h>` in X/Open so that the following macros will be visible: `htonl`, `htons`, `ntohl`, `ntohs`, `inet_addr`, `inet_lnaof`, `inet_makeaddr`, `inet_netof`, `inet_network`, and `inet_ntoa`.

**<netdb.h>**

`<netdb.h>` will include `<netinet/in.h>` in X/Open, so that the `in_addr_t` definition will be visible.

`h_errno` will be declared as `extern int`.

### <netinet/in.h>

- Two new types, `in_port_t` and `in_addr_t`, are defined. `in_port_t` is defined as an unsigned integral type of exactly 16 bits. `in_addr_t` is defined as an unsigned integral type of exactly 32 bits.
- The `sa_family` field in `sockaddr_in` is defined as `sa_family_t`. `sa_family_t` is implementation-specific. For HP-UX, `sa_family_t` is unsigned short, as in 10.01.

### Impact

If you use HP-UX Sockets, you will not see any impact.

If you are porting your applications to X/Open Sockets, you will have to make a few changes in your programming environments and in your source files. Refer to the above sections for more details.

### Compatibility

Applications using HP-UX Sockets for 10.10 will be fully compatible both in binary and source level. However, you are advised to migrate your applications to conform to X/Open Sockets. At some future release, any feature in HP-UX Sockets that is incompatible with X/Open Sockets will be obsoleted.

If you are porting your applications to X/Open Sockets, be aware of the few differences between HP-UX Sockets and X/Open Sockets. Refer to the above sections for more details.

### Size Requirement

Impact on memory usage will be insignificant. A new socket timer structure will be dynamically allocated if `SO_LINGER` option is used. The socket timer structure has a size of 16 bytes.

A new shared library, `libxnet.sl`, is added to support X/Open Sockets and XTI. The library has a size of about 98300 bytes.

---

## Streams

### Default MP Synchronization Change

*For 10.10:*

The default MP synchronization level of STREAMS modules and drivers is changed to the “queue” level synchronization Release 10.10. For a description of the MP synchronization level, see “Writing MP Scalable Modules and Drivers” in the *STREAMS/UX for the HP 9000 Reference Manual* (J2237-90005).

### Extended Signals

*For 10.10:*

To provide information about the event and band associated with a SIGPOLL signal, STREAMS supports extended signals (see the `siginfo(5)` manpage for details.) For the STREAMS events, special codes are defined in `<siginfo.h>`; the codes describe the reason why the SIGPOLL was generated. The `siginfo_t` structure is filled up as shown below and passed to the signal handler.

event	si_signo	si_code	si_band	si_errno
S_INPUT	SIGPOLL	POLL_IN	band readable	unused
S_OUTPUT	SIGPOLL	POLL_OUT	band writable	unused
S_MSG	SIGPOLL	POLL_MSG	band signaled	unused
S_ERROR	SIGPOLL	POLL_ERR	unused	Stream error
S_HANGUP	SIGPOLL	POLL_HUP	unused	unused
S_HIPRI	SIGPOLL	POLL_PRI	unused	unused

The `si_errno` returned for the S\_ERROR event is determined by the data contained in the M\_ERROR message that caused it. If the M\_ERROR message contains one byte of data, that data is used for the `si_errno`. When the M\_ERROR message contains two bytes of data (that is, one errno number for read-side and another for write-side) and if both errno's are neither zero nor



NOERROR, the write-side errno is returned. Otherwise, the read-side errno is returned.

## **XTI Internet Protocol-Specific Behavior of Urgent Data**

*For 10.10:*

XPG4 XTI version 2 specifies Internet protocol-specific behavior. XTI uses the T\_EXPEDITED flag to notify the application of the TCP urgent mode. As of 10.10, the behavior of XTI is supported.

When an M\_PCSIG message with SIGURG as its intended signal is sent upstream from any transport provider, XTI interprets the message as the notification of the beginning of the urgent mode. When an M\_PROTO with T\_EXPEDITED\_IND primitive is sent upstream from the transport provider while in urgent mode, XTI interprets that T\_EXPEDITED\_IND as the notification of the end of the urgent mode and interprets the data accompanied with the T\_EXPEDITED\_IND indications as the last part of the urgent data.

## **XTI/TLI and TIMOD Use**

*For 10.10:*

At 10.10, the XTI and TLI libraries no longer require the STREAMS TIMOD module be pushed onto the stack. Therefore, all applications compiled at 10.10 that have code expecting this module will need to remove this code. This should only affect applications that use the TIRDWR module for read/write semantics over a transport endpoint because they normally “pop” the TIMOD module off before “pushing” the TIRDWR module onto the stream stack.

This does *not* affect binary compatibility . Any application that currently relies on this module being present when a `t_open(3)` is performed and was compiled on a previous version of HP-UX, will still have that module pushed onto the stream stack.

---

<b>Note</b>	The TLI and XTI specifications do not specify TIMOD because this would be a library implementation issue and not a API issue.
-------------	---

---

## **Streams PTY**

*For 10.10:*

New code to the Streams PTY driver was added to recognize HP15 character sets and process them as if they are EUC characters. The `euccset(1)` command now allows you to enable HP15 character processing mode on the Streams PTY driver. The HP15 mode is mutually exclusive with the normal EUC mode.

Applications that uses HP15 characters now can use the Streams PTY. There is no impact to those applications that do not use HP15 characters. By default, the HP15 character processing mode is not turned on.

The HP15 character sets are primarily used by Asian NLS applications prior to the introduction of the EUC characters.

There is binary compatibility with previous version of the Streams PTY driver. The HP15 mode is mutual exclusive with the normal EUC mode and require an explicit command to turn it on.

## **LLIST ioctl**

*For 10.10:*

The `LLIST ioctl` has been changed to comply with SVR4.2. On success, the return value is zero. `sl_modlist` contains the list of module names and `sl_nmods` indicates the number of entries that have been filled in.

## **timod for XTI/TLI Libraries**

*For 10.10:*

The XTI/TLI libraries associated with 10.10 no longer require the `timod STREAMS` module be pushed on top of a transport provider. The functionality that the `timod` module provides has been incorporated into the `STREAMS/UX` framework.

Applications that have been created using an archived version of the libraries will run on 10.10 because `timod` still continues to be a default `STREAMS` kernel module.

## Streams Functionality

*For 10.0:*

Streams/UX became part of the base HP-UX product instead of being an independent, separately orderable product. Streams/UX consists of two filesets, STREAMS-KRN and STREAMS-RUN, which are provided as part of the base HP-UX operating system product. 10.x Streams/UX provides the same functionality delivered in the 9.0 Streams/UX product, and includes the following new functionality that was introduced at 10.0.

- Multi-processor performance scaling of STREAMS modules and drivers has been added.
- Support for a STREAMS-based pseudo-terminal (`pty`) subsystem has been added. Note that the STREAMS filesets themselves do not *supply* a STREAMS-based `pty`; they merely *support* one. Additional filesets must be loaded to implement a STREAMS-based `pty`.
- A generic XTI interface (XTI library and supporting STREAMS module) has been added. This interface is intended to be used in conjunction with user-supplied STREAMS-based transport provider code which can process TPI messages.
- The generic TLI interface provided as part of 9.0 Streams/UX has been modified to use new error codes as defined by SVID3.
- Capability has been added to allow creation of threaded applications that make STREAMS, TLI, or XTI calls.
- Streams/UX can be configured into a 10.x system using SAM.
- The `poll(2)` system works with either STREAMS or non-STREAMS file descriptors on both the 10.x Series 700 and 800.
- STREAMS tracing and logging has been integrated with the HP-UX Network Tracing and Logging product (NetTL). If NetTL is running, STREAMS error and trace messages generated by `strlog(7)` or `putmsg(2)` can be delivered to the NetTL subsystem in addition to the STREAMS trace/log facilities. This allows NetTL facilities to be used on STREAMS trace/log data and allows STREAMS trace/log data to be treated in a consistent manner with data from networking products that support NetTL.

- STREAMS system calls are part of 10.x `libc`, so it is no longer necessary to link with the `libstr.a` library (which is provided as part of the 9.0 Streams/UX product) when STREAMS applications are created.
- Major changes have been made to how HP-UX I/O drivers in general (including STREAMS drivers and to a lesser extent STREAMS modules) are configured and initialized in the HP-UX I/O system. Some of changes that are relevant to suppliers of STREAMS drivers and modules are as follows:
  - Restructuring of the master file into separate subsystem-specific sections.
  - Revision of the master file content to add new fields and entries, some of which are for configuration of MP synchronization levels for STREAMS modules and drivers or other STREAMS-related information.
  - Removal of some previously-reserved major number ranges, including the major number range formerly reserved for user-written STREAMS drivers (136-145), and the introduction of a new dynamic major number allocation facility for HP-UX I/O drivers.
- The behavior of `select(2)` has been slightly modified. The most significant change is that `select` returns an exception condition for a stream if a banded message or a high-priority message is at the head of the stream read queue.
- Modules (including drivers) that run in the uniprocessor emulation mode on multiprocessor machines, should use the “queue” MP synchronization level. (For information on the MP synchronization level, see “Writing MP Scalable Modules and Drivers” in the *Streams/UX for the HP 9000 Reference Manual* (J2237-90005)). If the “queue pair” synchronization level (which is the default value in the 10.01 release) is used instead, not only the module’s write-side put routine, but also the read-side put routines, may run on Interrupt Control Stack (ICS). HP recommends that developers of UP emulation modules set the “queue” level synchronization. The default synchronization level is changed to the “queue” level in HP-UX 10.10.

For more information, refer to the *Streams/UX for the HP 9000 Reference Manual* (J2237-90005).

## STREAMS-Based Pipes

*For 10.0:*

STREAMS-based pipes are fully backward compatible with the 9.x HP-UX pipe implementation and are fully POSIX compliant. However, you can choose to use the existing HP-UX pipe implementation instead. STREAMS-based pipes must be explicitly generated into the kernel.

The following is a summary of major changes that were made to STREAMS-based pipes at 10.0. Refer to *Streams/UX for the HP 9000 Reference Manual* (J2237-90005) for more information.

- At kernel generation time, you can specify whether you want all pipes that are created by the `pipe(2)` system call to be STREAMS-based.
- By installing the STREAMS-KRN fileset and including the appropriate STREAMS drivers, modules, and tunable parameters in the kernel system file, you can cause all pipes that are created by `pipe` to be STREAMS-based as described in the SVID3 `pipe(BA_0S)` manpage.
- The following calls are supplied: `fattach(3c)`, `fdetach(3c)`, `fdetach(1M)`, and `isastream(3c)`. These calls are used to associate/disassociate STREAMS or STREAMS-based pipes with an existing node in the filename space.
- STREAMS-based pipes may not be as fast as the HP-UX pipe implementation because the additional functionality and flexibility that they provide require more code.

---

## TERMIO

The terminal I/O drivers were modified at 10.0 to allow execution within driver code by multiple processors simultaneously. Access is limited to a single CPU for a single port (`pty` master/slave pair or MUX port), but can be concurrent between ports. Only MUX and `pty` drivers have been multi-threaded.

For SVID3 compliance, the `tcgetsid()` system call has been added and the `stty` command has been enhanced to allow the system-wide setting of the special character defaults used when a port is opened. Additional SVID3

functionality has been added to the `termio` line discipline and drivers. For more information, refer to the `tcgetsid(3c)`, `termio(7)`, and `stty(1)` manpages.

The `tty` device file naming-convention for Series 700 on-board serial ports and call-in and call-out modem device files has changed from `tty<xx>` to `tty<x>p0`. Refer to the `insf(1M)` manpage for more information. The `tty` device file minor number formats have changed. Refer to the `termio(7)` manpage. In addition, for MUX drivers, the device file naming-convention and the method for allocating major numbers have changed. Refer to the “I/O Convergence” section earlier in this document and the *Configuring HP-UX for Peripherals* manual for more information.

A STREAMS `pty` driver is provided on HP-UX 10.x. However, it does *not* replace the 9.x BSD-style `pty` driver, which is still supported on HP-UX 10.x.

A new kernel tunable parameter, `nclist`, has been added to allow the number of character blocks in the system to be tuned independently of `MAXUSERS`. In pre-10.0 releases, you had to change `MAXUSERS` to change the number of cblocks in the system. Since the default value of `nclist` is  $(100 + 16 * \text{MAXUSERS})$ , if you do not specify a value for `nclist`, the `MAXUSERS` value will control the number of cblocks. This is a convenient way of sizing `nclist`. However, if you want to tune `nclist` without changing `MAXUSERS` and the other tunables calculated from `MAXUSERS`, this is now possible.

---

## XTI

*For 10.10:*

Changes to XTI are made to conform to X/Open XTI CAE Specification Version 2. When a TCP urgent byte is received by the transport, all data currently buffered in the kernel for this connection will have the `T_EXPEDITED` flag set when read. If there are multiple data buffers, each will also have the `T_MORE` flag set, except the last buffer whose last byte will be the urgent byte. Refer to the X/OPEN Transport Interface (XTI) Version 2, Appendix B.3 under the `t_rcv()` function for more details.

## Commands and Libraries

---

This chapter provides summary information about commands and library routines that are new or have changed and are not discussed in other parts of this document. Unless specified otherwise, the changes occurred at 10.0.

This chapter covers the following topics:

- Accounting Commands
- **Changed Commands in Section 1**—*changed at 10.20 and 10.10*
- **Changed Commands in Section 1M**—*changed at 10.20 and 10.10*
- Commands No Longer Supported
- **Library Routines (libc)**—*changed at 10.20 and 10.10*
- Math Library Routines (libm, libM)
- Changes to Other Commands and Files—*changed at 10.10*

---

**Note**

- Because of the extensive changes to some commands, HP is also supplying their old (9.x) versions. These versions will be stored in the directory `/usr/old`. HP does not guarantee to continue to support the old versions in future releases.
  - HP is providing a mechanism to link 10.x commands to their old (9.x) locations. These links are in effect by default on all 10.x systems, so references to the 9.x pathnames in code and scripts will find the commands on a 10.x system. See the *Upgrading from HP-UX 9.x to 10.x* manual for more information.
-

---

## Accounting Commands

*For 10.0:*

- Commands and files moved to conform to 10.0 file system layout (see “File System Layout Changes”).
- Comply with SVID3, SVR4.
- Changes to `runacct(1M)` arguments (see “Changed Command `runacct(1M)`”).

### File System Layout Changes

Type of file	Old location	New location/ <i>(Comments)</i>
=====		
Data	<code>/usr/adm/acct</code>	<code>/var/adm/acct</code>
Commands	<code>/usr/lib/acct</code>	<code>/usr/sbin/acct</code>
(Holidays)	<code>/usr/lib/acct/holidays</code>	<code>/etc/acct/holidays</code>
Startup	<code>/etc/rc</code>	<code>/etc/rc.config.d/acct</code> <i>(Set <code>START_ACCT</code> to 1.)</i>

- To start up system accounting, set `START_ACCT` to 1 in `/etc/rc.config.d/acct`.
- To invoke the accounting commands, use the path `/usr/sbin/acct/command`. (See Note below.)
- To see the output of the commands, look in the directory `/var/adm/acct`.

See “The HP-UX 10.0 File System Layout” in Chapter 4 for more information on the new file system layout.

### Changed Command `runacct(1M)`

The `CONNECT1` and `CONNECT2` arguments to `runacct` have been replaced by `CONNECT`. The `CONNECT` entry point is identical to the old `CONNECT1` entry point.



---

## Changed Commands in Section 1

This section summarizes changes to commands in Section 1 of the *HP-UX Reference* manual (and the online manpages) that are not discussed elsewhere in this document. Unless specified otherwise, the changes occurred at 10.0.

Commands changed at 10.20:

- `cancel(1)`
- `cpio(1)`
- `ex(1)`
- `findmsg(1)`
- `ftio(1)`
- `iconv(1)`
- `lp(1)`
- `lpstat(1)`
- `pax(1)`
- `tar(1)`
- `vi(1)`

Commands changed at 10.10:

- `admin(1)`
- `ar(1)`
- `asa(1)`
- `at(1)`
- `bc(1)`
- `cal(1)`
- `cancel(1)`
- `cksum(1)`
- `compress(1)`
- `crontab(1)`
- `cs(1)`
- `ctags(1)`
- `cu(1)`
- `date(1)`
- `dd(1)`
- `diff(1)`
- `du(1)`
- `eucset(1)`

- `ex/vi(1)`
- `expand(1)`
- `expr(1)`
- `find(1)`
- `ftio(1)`
- `getconf(1)`
- `grep(1)`, `egrep(1)`, `fgrep(1)`
- `id(1)`
- `initstate()`
- `ksh(1)`
- `lex(1)`
- `lpstat(1)`
- `ls(1)`
- `mail(1)`
- `make(1)`
- `man(1)`
- `mkfifo(1)`
- `nice(1)`
- `nl(1)`
- `nohup(1)`
- `od(1)`
- `pack(1)`
- `patch(1)`
- `pathchk(1)`
- `pax(1)`
- `pr(1)`
- `prs(1)`
- `ps(1)`
- `renice(1)`
- `rm-del(1)`
- `sact(1)`
- `sccs(1)`
- `sed(1)`
- `sh-posix(1)`
- `split(1)`
- `strings(1)`
- `strip(1)`
- `tar(1)`

#### **7-4 Commands and Libraries**

- time(1)
- top(1)
- tput(1)
- tsort(1)
- uncompress(1)
- unexpand(1)
- uniq(1)
- unpack(1)
- uuencode(1)
- uulog(1)
- uupick(1)
- uustat(1)
- uuto(1)
- uux(1)
- val(1)
- vi(1)
- wc(1)
- what(1)
- who(1)
- xargs(1)
- zcat(1)

Commands changed at 10.01:

- cu(1)
- man(1)
- uucp(1)

Commands changed at 10.0:

- adjust(1)
- awk(1)
- cat(1)
- chown(1), chgrp(1)
- cp(1)
- cpio(1)
- doschmod(1), etc.
- echo(1)
- ftio(1)
- gencat(1)

- help(1), sccshelp(1)
- iconv(1)
- iostat(1)
- join(1)
- locale(1)
- lp(1), lpadmin(1)
- mesg(1)
- mkdir(1)
- mm(1)
- more(1)
- mt(1)
- mv(1)
- nljust(1)
- nroff(1)
- passwd (1)
- paste (1)
- rm(1)
- rmdir(1)
- sar(1)
- sort(1)
- sum(1)
- tar(1)
- tr(1)

## **adjust(1)**

Modified to support Japanese line-breaking rules:

- No line will start with a *bkinsoku* character
- No line will end with a *ekinsoku* character

One or other of these rules will be broken if the line cannot be ended any other way.

The japanese locale has two new char classes, **bkinsoku** and **ekinsoku**.

## admin(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- When “-” is specified as an argument, SCCS directory names given on standard input are processed (they were ignored on earlier releases).
- Unreadable files are now silently ignored.

## ar(1)

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## asa(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- The output from `asa` depends on the first character removed from the input line:

When the first character is:	asa outputs:
a blank space	the rest of the input line without changes
0	a newline character followed by the rest of the input line
1	one or more characters that causes an advance to the next page, followed by the rest of the input line.
+	asa replaces the newline character of the previous line with one or more implementation-dependent

characters that causes printing to return to column position 1 followed by the rest of the input line.

(no file operands          standard input  
are specified)

- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- **asa** processes operands in the order specified on the command line.
- When an input file is defined as a text file, at least 2048 bytes can be accumulated from a set of continued input lines.
- When no error occurs during the execution of a utility, no error messages are written to standard error and the exit status from the utility is zero.
- When a utility is unable to perform the requested action on an external object (such as a file, directory, user, process, and so on) specified by an operand, the utility issues a diagnostic message to standard error and continues processing subsequent operands. The final exit status of the utility is non-zero.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

## **at(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- The “Usage” message has been changed to reflect the availability of the **t** option.
- Option arguments for the **-f** and **-q** options are mandatory.
- The **abday** keyword is accepted.

See the **at(1)** manpage for details and syntax.

## **awk(1)**

*For 10.0:*

- Variable assignment
- **srand()**
- FS and newline characters
- Backslash character

## **7-8 Commands and Libraries**

Variable assignment      `awk` will make the variable assignment just before processing the file operand that follows the variable, which will not be visible for the rest of the unprocessed files.

Uninitialized variables and strings will be printed as empty strings.

`srand()`      The arithmetic built-in function `srand()` returns the previous seed value.

FS and newline characters      The FS character, and the newline character followed by the last field, are no longer considered in the computation of NF.

backslash character      Within strings defined by double quotes, the backslash character (`\`) serves as an escape character to introduce escape sequences, as shown below.

Supported sequences introduced by the backslash are:

Escape Sequence	Meaning/Representation
=====	
<code>\a</code>	<alert>
<code>\b</code>	<backspace>
<code>\f</code>	<form-feed>
<code>\n</code>	<new-line>
<code>\r</code>	<carriage-return>
<code>\t</code>	<tab>
<code>\v</code>	<vertical-tab>
<code>\ddd</code>	The character whose encoding is represented by a one-, two-, or three-digit octal integer.
<code>\&lt;char&gt;</code>	(where <char> is any character not specified in this table): the character <char>.

In previous releases the sequence `\<char>` was interpreted literally (for example, `\e` in 9.x was interpreted as `\e`; whereas in 10.x, `\e` is interpreted as `e`).

This change brings HP's treatment of the backslash into conformity with other implementations of `awk`.

## **bc(1)**

*For 10.10:*

- Very large numbers will be split across lines with 70 characters (this includes \ and \n characters).
- `scale(zero)` is 0 even though `bc` is invoked with the `-l` option.

## **cal(1)**

*For 10.10:*

- Now handles different international locales.
- The output of `cal(1)` has changed:
  - The column width of a multi-byte character can be at most 4. At least one multi-byte character will be printed.
  - It is only possible to format two months per row in the calendar.
  - The column width of a day has increased from 3 to 5.
- The internationalized environment variables, such as `LANG` are now supported.
- The day and month names have been replaced by corresponding abbreviated day and month versions of `nl_langinfo` items.
- Hard-coded values have been changed to `#defined` values whenever possible.
- Error messages are written to the standard error file (`stderr`).

## **cancel(1)**

*For 10.20:*

These model script files are added for new printers:

- `deskjet850C`
- `deskjet855C`
- `deskjet1600CM`

You will be able to select these model scripts for configuring your printers.

The `cancel(1)` command is enhanced to interoperate with the new HP Distributed Print Services (HPDPS), a separate print spooling system. The command can submit jobs to the HPDPS, cancel HPDPS jobs, and list HPDPS printers and jobs.

## **7-10 Commands and Libraries**



*For 10.10:*

- Now conforms to XPG4 specifications.
- `cancel(1)` can be localized; a code for internationalization has been added.
- Error messages are written to the standard error file (`stderr`).
- Whenever an error occurs, the exit status is non-zero.

## **cat(1)**

New options:

- `-b` Omits line numbers from blank lines; `-n` option is assumed.
- `-n` Displays lines preceded by line numbers, starting with 1.
- `-r` Replaces multiple consecutive empty lines with one empty line.

Changed options:

The options `-e` and `-t` no longer require `-v`.

## **chown(1), chgrp(1)**

These commands now behave the same way (following SVID3) on the Series 700 and 800 when they encounter symbolic links. The default action is to follow the link.

New option:

- `-h` Do not follow the symbolic link.

The `hpux_aes_override` kernel flag is no longer needed.

## **cksum(1)**

*For 10.10:*

- “`--`” as an argument indicates the end of options and the beginning of operands. Any arguments following the “`--`” that starts with a “`-`” are considered to be operands (not options).

*For 10.0:*

Checksum values are generated according to a new algorithm as of 10.0 to ensure conformance with the latest POSIX.2 standard.

This means that checksum values will differ between 9.x and 10.x for the same input. If you need to obtain checksum values calculated by the 9.x algorithm, you can use the old version of the command, delivered with 10.x as `/usr/old/usr/bin/cksum`. If you are using these values in a script, use `/usr/old/usr/bin/cksum` to verify them, then update them to the 10.x values using the 10.x command.

`cksum` now also supports piped input.

## **compress (1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- If no file is specified or if the file operand is “-”, standard input is compressed to standard output.
- Returns a value of 1 whenever it fails on one of the operands (previous versions returned a value of 0).
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

*For 10.0:*

- New option `-z`:
  - z is same as `-f` except that it does not force compression when there is null compression.

## **cp(1)**

`cp` will provide warning messages whenever extent attributes of a file are lost. You can use the `-e` option to control what `cp` does if it cannot copy extent attributes.

New option: `-e`

This option can have one of the following three parameters:

```
-e [warn | force | ignore]
```

`-e warn` is the default behavior of the command.

## **7-12 Commands and Libraries**

If the extent attributes of a file cannot be copied, then:

- If **-e warn** is specified, or if the **-e** option is not used, **cp** will issue a warning message and copy the file.
- If **-e force** is specified, **cp** will not copy the file.
- If **-e ignore** is specified, **cp** will ignore extent attributes.

See also entries for “**cpio(1)**”, “**ftio(1)**”, and “**mv(1)**” in this section.

## **cpio(1)**

*For 10.20:*

Because of industry standards and interoperability goals, files larger than 2GB cannot be archived using **cpio(1)**. You will receive an error message if archival is attempted for a file that is larger than 2GB in size.

User and group IDs larger than 60K are not supported. Files with user/group IDs greater than 60K are archived and restored under the user/group IDs of the current process.

The **fbackup** and **frecover** command set is recommended for backing up files that are larger than 2GB or are owned by user/group IDs that are greater than or equal to 60000.

*For 10.0:*

**cpio** will provide warning messages whenever extent attributes of a file are lost. You can use the **-e** option to control what **cpio** does if it cannot copy extent attributes.

**cpio** will not allow files from pre-10.0 systems to be restored onto a 10.x system because the device files are different and some of them are invalid for 10.x.

See the entry “**cp(1)**” in this document for more information; for full details, see the **cpio(1)** manpage.

## **crontab(1)**

*For 10.10:*

- A new `-e` option has been added. When specified, `-e` edits a copy of your `crontab` file or creates an empty file to edit if the `crontab` file does not exist. When editing is complete, the file is copied into the `crontab` directory as your `crontab` file.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **csch(1)**

*For 10.10:*

- The `limit` built-in command is now available. This command allows you to access and/or modify system resources for the following: `cputime`, `filesize`, `datasize`, `stacksize`, `coredumpsize`, and file descriptors.

*For 10.01:*

- `exit` in `.cshrc` now works correctly.
- `set nonomatch` now always works correctly.
- `foreach var (wordlist)` syntax now allows alphanumeric variables in the `var` field.
- `csch` no longer hangs if `SIGUSR1` is ignored and `csch` is `exec`'ed.
- `LINES` and `COLUMNS` are set by default when `csch` starts up.

See “Shells” in Chapter 5 for more information on HP-UX shells for 10.x.

## **ctags(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **7-14 Commands and Libraries**

- The `-f` option for specifying tags-file name has been added. When `-f` is not specified, the output *tags-file-name* is `tags`.
- The actions of the `-t` option are now the defaults; the `-t` option has been made redundant for backward compatibility.
- The usage message has been changed to add the `-f` option.
- If an error occurs when processing one of the operands, the final exit value has been changed to be non-zero.
- The global variable `lineftell` has been fixed. `lineftell` is now initialized to zero for each file.

## **cu(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The `-d` level option has been renamed to a `-D` level. This option prints diagnostic traces. The level specified can be from 0 to 9, where high levels produce more detail in the diagnostic messages.
- The `-d` option has been added to write diagnostic traces. This option is equivalent to a `-D 9` option.
- The usage message has been modified to reflect the above changes.
- If the `-n` option is used, a message prompting for the telephone number is delayed until the command line processing is complete.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

*For 10.01:*

As a consequence of the new NFS file-sharing paradigm, the `uucp(1)` and `cu(1)` commands are now supported on client systems.

This change applies to all commands in the `UUCP` fileset, as well as the commands in the `SYSCOM` fileset:

```
cu
ct
uupath
kermit
umodem
vt
vtdaemon
vtgateway
```

`vtserver`

Refer to the section in this chapter on the `uucp(1)` command for additional information.

### **date(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- No response is required from the user if the date is being set backward.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

### **dd(1)**

*For 10.10:*

- `dd(1)` can now copy disk devices larger than 4Gb.

### **diff(1)**

*For 10.10:*

- When the command fails, a return value will be non-zero.

### **doschmod(1), etc.**

The following commands support wildcard characters in MSDOS file names:

- `doschmod(1)`
- `doscp(1)`
- `dosls(1)`
- `dosrm(1)`
- `dosrmd(1)`

You can use the metacharacters `*`, `?`, and `[ ... ]` when specifying DOS file names. You must enclose them in quotes, because file name expansion must be performed by the DOS utilities, not by the shell. DOS utilities expand file names as described in the `regexp(5)` manpage under “PATTERN MATCHING NOTATION.”

## **du(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- A new option **k** has been added, which give the file size in units of 1024 bytes, rather than the default 512-byte unit.
- Option **r** is the default.
- Options **s** and **a** are mutually exclusive. If both are specified, a usage message will be displayed.
- If an error occurs, a final exit value has been changed to be non-zero.

*For 10.0:*

Now supports multiple file system types.

Enhanced option:

`-t [type]` now works with all supported file system types.

## **echo(1)**

Recognizes two additional special characters:

```
\a      write an alert character
\0[num] write an 8-bit value representing the octal number [num]
```

## **eucset(1)**

*For 10.10:*

- A new option `-c codeset` has been added to support the HP-15 codesets. Options to `-c` can be one of the following, depending on the codeset:
  - BIG5
  - CCDC
  - GB
  - SJIS

This enables you to set the `cswidth` parameter for HP-15 codesets. You do not have to explicitly specify the `cs` width parameter (it is derived from the locale).

## **ex/vi(1)**

*Changes at 10.20 for ex(1):*

- Now conforms to XPG4 specifications.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.
- If the implementation supports the `ctags` utility, when `-t tagstring` is specified with either `-c` or `+command`, `ex` processes the `-t tagstring` first and then executes the command.
- The `ex` command does not read the `.exrc` file unless it is owned by the same user ID as the effective user ID of the invoking process.
- When autowrite is set and the current buffer has been modified, a `next`, `rewind`, `tag`, `edit`, `suspend`, or `stop` command not followed by a `!` or a `!` command causes the buffer to be written to the current file.
- The exit status must be non-zero when an error or warning message is encountered. The error or warning message must be written to `stderr`.
- When the range change `count` command is issued, `ex` enters input mode and replaces the first line specified by range and the next `count -1` lines with the text entered. The current line is set to the last line entered.
- When autowrite is set and the current buffer has been modified, a `next`, `rewind`, `tag`, `edit`, `suspend`, or `stop` command followed by a `!` does not cause the buffer to be written to the current file. The command checks if the `!` is also given with the `next` command. If it is, the file should not write into the next file even if “autowrite” is set.
- When the `range` move line command is specified, `ex` moves the lines specified by `range` (the default is the current line) to be after line. The current line indicator is set to the first of the moved lines.
- When the unnamed buffer contains lines from a previous delete or yank command, `line put` places the lines in the unnamed buffer after line (the default is the current line) and sets the current line indicator to the first line put back.
- When the `SHELL` environment variable is set to “myshell” and the read `!file` command is invoked, the command `myshell -c file` is called and the



standard output of this command is placed in the current buffer after the current line.

- When **beautify** or **bf** is set and a command is given to read a file into the buffer and the file contains non-printable characters other than tab, newline or form-feed, those non-printable characters are discarded from the text read in.
- All diagnostic and error messages are written to standard error.
- When the **edcompatible** set option is enabled and a **substitute** command is entered with either the **c** or **g** options, all subsequent **substitute** commands entered without the specification of the **g** or **c** options (respectively) are executed as if the appropriate option had been specified.
- When a regular expression contains the character sequence **\>**, this sequence is matched against the beginning of a word in the lines being matched.
- When the **range &** command is invoked, the substitution pair **/pattern/repl/** from the previous substitute is used and the first instance of a string that matches the regular expression **pattern** is replaced by the string **repl** on each line specified by **range** (the default is the current line.)
- When a RE is **[a-z]\*\$**, **ex** fails to find the match and does not perform the substitution. This is due to an anchor problem “\$”.

These changes will make the **ex** command binary incompatible. However, the incompatible behavior will only be seen when the **UNIX95** environment variable is set.

*Changes at 10.10 for ex(1):*

- Option **-c** has been added. This option is the same as the **+** option.
- Option **-s** has been added. This option is the same as the **-** option.

*Tag Stacking Feature for 10.0:*

The **:tag** command now includes a tag stacking feature. The new **:pop** command will remove a tag from the tagstack and position the editor to the previous tag location. Up to 30 tags can be stored in the tagstack.

In **vi**, the **[Ctrl-T]** command will pop a tag and go to the previous tag location. It is analogous to the existing **[Ctrl-]** command, but can be used repeatedly to change tag locations until the tagstack is exhausted.

A new edit option (`tagstack`, `tgst`) enables this feature. The default setting is on. If `notagstack` is selected, the tags feature will behave as it did before 10.0.

## Regular Expressions

Before 10.0, `ex/vi` used its own regular expression handler. It now uses the standard `regcomp(3C)` and `regex(3C)` routines. You will notice the following changes:

- The following constructs are now supported.

```
\{m\}
\{m,\}
\{m,n\}
```

- There are restrictions affecting beginning-of-word and end-of-word:

- Beginning-of-word can be used only at the beginning of a regular expression, not in the middle. For example,

```
:s/\<word/new
```

is valid, but

```
:s/word1 \<word2/new1 new2
```

is not.

- Similarly end-of-word can be used at the end of a regular expression, but not in the middle. For example,

```
:s/word\>/new
```

is valid, but

```
:s/word1\> word2/new1 new2
```

is not.

- Beginning and end of word constructs will not recognize an ASCII-multibyte character transition as a word boundary.

For example, if `xx` is a multibyte character:

```
xxAxx5xxYS
```

## 7-20 Commands and Libraries

A, 5, YS, and the three **xx** characters would each be distinguished as words by **vi**'s **w** command. But the search

```
/\<A
```

would not find the word beginning with **A** because it is not delineated by a multibyte character.

### **expand(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The range of **tabstop** changed from 0 through 256 to 0 through 2147483647.

### **expr(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- When two strings are given as arguments and do not match, a new line is not output to the standard output.
- **expr** suppresses leading zeros. For example, 000001 is output as 1.
- **expr** supports nesting of parentheses to any depth.

### **find(1)**

*For 10.10:*

A period (.) in a filename is matched by using a period as the first character or immediately following a slash character in the pattern.

*For 10.0:*

Recognizes two additional special characters:

```
\a      write an alert character
\0[num] write an 8-bit value representing the octal number [num]
```

Now supports multiple file system types.

Enhanced options:

- fsonly [type] now works with all supported file system types.
- fstype [type] now works with all supported file system types.

## findmsg(1)

*For 10.20:*

findmsg(1) supports print specifiers, such as PRI\* defines in `inttypes.h`, as part of a message within a `catgets()` format. It also supports `#ifdefs` to select the specific messages from the input file.

For example, the message for

```
printf(catgets(catd, NL_SETN, mno, "msg1 " PRIuMAX" \n"));
```

could be read as

```
"msg1 <value_of_PRIuMAX> \n"
```

Input file(s) will be preprocessed using the C preprocessor (`cpp(1)`) to achieve the above features. The preprocessor recognizes the following new options:

- D<sym> Defines the symbol <sym> for `#ifdef`.
- U<sym> Undefine the symbol <sym>
- i Using this option, you can only extract the required messages under `#ifdefs`. That is, -D and -U are also used to select the text in the input file.

Without this option, -D and -U options are only used to select the print specifiers, not the messages from the input file.

- v This option outputs the errors issued by `cpp(1)`. By default, errors issued by `cpp` will not be displayed.

## Impact

If more than one string occurs in the message part of the `catgets()` format (`catgets(catd, NL_SETN, msgno, "str1" "str2")`), all strings will be concatenated as a single string (`catgets(catd, NL_SETN, msgno, "str1str2")`).

## Performance

Input files that do not contain print specifiers are preprocessed even when it is not required. This is an overhead for such files.

## ftio(1)

*For 10.20:*

`ftio(1)` creates archives that are compatible with `cpio(1)`. Because of industry standards and interoperability goals, `cpio(1)` does not support the archival of files larger than 2GB. User and group IDs larger than 60K are not supported. Consequently, `ftio(1)` exhibits the same behavior. Therefore, files larger than 2GB cannot be archived using `ftio(1)`. Files with user/group IDs greater than 60K are archived and restored under the user/group IDs of the current process.

## Compatibility

There are no compatibility issues for files less than 2GB in size. There is a new error message if archival is attempted for a file that is greater than 2GB in size.

## Alternatives

The `fbackup` and `frecover` command set is recommended for backing up files that are larger than 2GB or are owned by user/group IDs that are greater than or equal to 60,000.

*For 10.10:*

The `ftio(1)` command can now perform tape copy to a remote tape device. `ftio(1)` was modified to determine where the `rmt` command exists on the remote system.

If you encounter this problem, you can, on a 9.x system, copy the file `/etc/rmt` to `/usr/sbin/rmt`.

*For 10.01:*

`ftio` will provide warning messages whenever extent attributes of a file are lost. You can use the `-e` option to control what `ftio` does if it cannot archive extent attributes.

See “`cp(1)`” in this section for more information. “`cpio(1)`” and “`mv(1)`” also support `-e`.

## **gencat(1)**

New option:

- specifies `stdin` or `stdout` for the source or catalog files, respectively.

## **getconf(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Proper parse and evaluation of the expression for the argument variable is performed when necessary.

See the `getconf(1)` manpage for details.

## **grep(1), egrep(1), fgrep(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- `grep -l` with `stdin` as input will output a message on `stdout`. This also occurs for `-l -m`.
- A `grep` on regular expressions prints correct results and returns exit 0 status for a successful `grep` with `-e`, `-f`, `-i`, `-x`, and `-v` options.
- Null pattern selects every input line. This also occurs for the `-e`, `-f`, and `-v` options.
- Null pattern with the `-E` option selects every input line.

- `grep -F` and `fgrep` results match. The results also match using similar valid parameters applied to both.
- Using the `-q` option and the first inaccessible input file results in an exit with zero status.

*For 10.01:*

The collating order of characters has been changed in some locales. See the “Locale Changes” section in the *README for NLS in 10.01*.

### **help(1), sccshelp(1)**

The name of the `help` command has been changed to `sccshelp`, and the back-end command, `help2 (/usr/lib/help/lib/help2)`, is no longer supported as an independent command: its functions are now part of `sccshelp`.

The reason for the change is that the command is really part of SCCS (the Source Code Control System) and the help libraries only exist for SCCS commands (`admin(1)`, `delta(1)`, `get(1)`, `prs(1)`, etc.). You do not need to change help libraries that you have created.

### **iconv(1) and iconv(3)**

*For 10.20:*

The `iconv(1)` and `iconv(3)` commands now support the conversion of characters between some of the HP-supported code sets and various forms of the ISO 10646 code set, specifically the UCS forms, UCS-2, and UTF-8.

---

<b>Warning</b>	<b>Do not store UCS-2 data in files. If you do, unexpected results might occur. Only UTF-8 data can be safely stored in files for processing.</b>
----------------	---

---

<b>Warning</b>	<b>This is only a conversion feature to facilitate input/output with other systems and for code-internal purposes. No other HP-UX commands, utilities, services, and so on has been converted to support UCS-2 or UTF-8 at this time. Only <code>iconv</code> supports this feature.</b>
----------------	--

---

The following conversions are supported:

```
UTF-8, UCS-2 <--> roman8, iso8859-1 for West European languages
UTF-8, UCS-2 <--> SJIS, eucJP (both including UDC/VDC) for Japanese
UTF-8, UCS-2 <--> eucKR for Korean
UTF-8, UCS-2 <--> hp15CN for Simplified Chinese
UTF-8, UCS-2 <--> roc15, big5, eucTR for Traditional Chinese
UTF-8, UCS-2 <--> thai for Thai
UTF-8, UCS-2 <--> iso8859-2 for East European
UTF-8, UCS-2 <--> iso8859-5 for Cyrillic
UTF-8, UCS-2 <--> iso8859-6 for Arabic
UTF-8, UCS-2 <--> iso8859-7 for Greek
UTF-8, UCS-2 <--> iso8859-8 for Hebrew
UTF-8, UCS-2 <--> iso8859-9 for Turkish
UTF-8 <--> UCS-2
```

## Impact

Additional tables and methods have been added to support this conversion capability.

No current code or applications are impacted. This feature will enable HP-UX to exchange and interoperate with other systems that support ISO 10646 characters.

*For 10.10:*

- Now conforms to XPG4 specifications.
- Increased codeset conversion support.
- Warning messages will no longer be printed.
- New codeset naming convention, but old convention still supported.
- Shift-in characters not added while converting to IBM or JIS codesets, if an ASCII character is the first character of the record.

See the *README for NLS in 10.01* and the `iconv(1)` manpage for details.

## id(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- You do not have to be superuser to run `id otheruser`, where *otheruser* is another user name with a different user ID.

## 7-26 Commands and Libraries



## **iostat(1)**

The output format of `iostat` has been substantially revised to make it more readable. Scripts that depend on the 9.x output format will need to be recoded.

The `maps` output field always reads 1.0; changes in disk technology have rendered this field meaningless, and it may be removed altogether in a later release.

## **join(1)**

Option no longer supported:

`-a` without a number will now result in an error. As of 10.0, `-a` must be followed by 1 or 2.

## **locale(1)**

Changed options:

- `-a` Lists all available public locales.
- `-k` Displays the name of the `charmap` file if `charmap` is specified. Supports new keywords in the `LC_TIME` category.
- `-c` Supports the `CHARMAP` category.

## **ksh(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- New signals `SIGXCPU` and `SIGXFSZ` are available.

The pre-HP-UX 10.10 history file will be discarded. Users who use multiple versions of HP-UX that NFS mount the same directory that contains the history file (usually the home directory) may want to create a separate history file for each HP-UX version. To do this, set the proper `HISTFILE` parameter. See the `ksh(1)` manpage for recommendations about setting the `HISTFILE` parameter to avoid the history file discard problem.

## **lex(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Options `-v` and `-n` are mutually exclusive.
- “`--`” as an argument indicates the end of options and the beginning of operands. Any arguments following the “`--`” that starts with a “`-`” are considered to be operands (not options).
- If no file is specified or if the file operand is “`-`”, the standard input should be used.

## **lp(1), lpadmin(1)**

*Changes at 10.20 for lp(1):*

For 10.20, these model script files are added for new printers:

- `deskjet850C`
- `deskjet855C`
- `deskjet1600CM`

You will be able to select these model scripts for configuring your printers.

The `lp(1)` command is enhanced to interoperate with the new HP Distributed Print Services (HPDPS), a separate print spooling system. The command can submit jobs to the HPDPS, cancel HPDPS jobs, and list HPDPS printers and jobs.

*For 10.0:*

New model files:

```
deskjet500C
deskjet550C
deskjet1200C
laserjet4
laserjet4Si
paintjetXL300
```

These files will support PCL5 features such as language switching, 600 dpi, tray selection, media type, and page level recovery. A summary of the options will print out on the banner page.

## **7-28 Commands and Libraries**

## **lpstat(1)**

*For 10.20:*

These model script files are added for new printers:

- deskjet850C
- deskjet855C
- deskjet1600CM

You will be able to select these model scripts for configuring your printers.

The `lpstat(1)` command is enhanced to interoperate with the new HP Distributed Print Services (HPDPS), a separate print spooling system. The command can submit jobs to the HPDPS, cancel HPDPS jobs, and list HPDPS printers and jobs.

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- The exist status is non-zero whenever an error occurs.

## **ls(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The `-t` option has changed so that comparisons occur between a collating sequence if the operands are equal by modified time.
- On output, the first *newline* character has been removed when the first item is to be written is a directory.
- The XPG4 behavior is activated if you set the UNIX95 environment variable.

*For 10.01:*

`ls` can now display the extent attributes of a file.

New option:

- e            Displays extent attributes. Must be used with `-l`; if `-l` is not specified `-e` is silently ignored. `-e` causes `ls` to open and

immediately close each file in the list; this could make the command execute more slowly.

Other commands that have changed to handle extent attributes are `cp(1)`, `cpio(1)`, `ftio(1)`, and `mv(1)`.

## **m4(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Fixes were made to the `ifdef` built-in macro.
- Support of multiple occurrences of `m4wrap` built-in macro.
- An error is flagged when a non-numeric argument is passed to the following built-in macros: `decr`, `divert`, `eval`, `incr`, `m4exit`, `substr`, and `undivert`.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- When an error occurs while processing one of the operands, processing continues with remaining operands and the final exit status is non-zero.
- The `ifdef` macro now makes an additional check to see if the value of its first argument is ‘0’.
- When the `m4wrap` macro is used multiple times, only the latest `m4wrap` argument is processed on EOF.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

## **mail(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **make(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- If the target is up-to-date, **make** prints a message and exits with a zero status.
- If the target is not up-to-date and the target rule contains no associated commands, **make** writes a message to **stdout** indicating that no action was taken for that target and exits with a zero status.
- The command with **a** as a prerequisite for targets, where **a** exists and is newer than each existing target, writes a message to **stdout** indicating that no action was taken and exits with a status of zero.
- Precedence of the command-line options over the **MAKEFLAGS** environment.
- **MAKEFLAGS** in the **makefile** replaces the **MAKEFLAGS** environment.
- Macros defined on the command line are added to the **MAKEFLAGS** variable.
- Macros defined in the **MAKEFLAGS**, but without any command line macros, adds the macro to the environment, overwriting any variable of the same name.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

*For 10.0:*

- Now conforms to XPG4 specifications.
- Search sequence for the **makefile** has been extended to include the **SCCS** directory.

Set the environment variable **PROJECTDIR** to search for an **SCCS** subdirectory that is not under the current working directory.

- Two new default macros: **SCCSFLAGS** and **SCCSGETFLAGS**.
- New special macro **VPATH** enables **make** to search through a list of paths (separated by colons) looking for prerequisites and dependencies.
- Conditional macro definitions supported.
- **include** supported.

## **man(1)**

*For 10.01:*

**man** will no longer create **cat\*** subdirectories automatically when run on a diskless client.

The **man** directories for an NFS Diskless client are under **/usr/share/man**, which resides on the server in the read-only portion of the file system. This means that the **man** command, when run from a client, will not install the formatted entry into the **cat\*** directories, because only the server can write to them.

You may want to run **catman(1M)** on the server to set up the **cat** directories. Otherwise, users will get the message,

Reformatting entry. Wait ...

and will have to wait for the manpage to be uncompressed, every time they run **man**.

See “**catman(1M)**” in this section or the **catman(1M)** manpage for information on running **catman(1M)** on a server that does not have common manpages with a given client.

On servers and standalone systems, the **man** command will continue to install a formatted entry in the **cat\*** directory.

## **mesg(1)**

A new option, **g**, sets access mode to 620.

This is the default and recommended access mode for **tty/pty** lines, allowing legitimate commands which are secure (e.g., **write(1)**) to be used by other users for sending messages.

## **mkdir(1)**

On 9.0 Series 800s, **mkdir -p** could go across a read-only file system and follow a soft link to a read-write file system to make a directory. **mkdir -p** does not have this capability on any 10.x computers.

## **mkfifo(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- If the mode specified by the user is **+**, **-**, or **=** without user (**u**), group (**g**), other (**o**), or all (**a**) specified, the mode bits gets set or cleared for all.
- If the mode specified by the user has the operator **+** with user (**u**), group (**g**), other (**o**), or all (**a**), the permissions for the newly created directory will be permissions specified after the operator and also the permissions as specified in the **umask**.
- If the mode specified at the command line has the operator with user (**u**), group (**g**), other (**o**), or all (**a**), the permissions for the newly created directory will be the permissions as per the **umask** value minus the ones specified.
- If the mode specified at the command line has the operator **=** with user (**u**), group (**g**), other (**o**), or all (**a**), the permissions for the newly created directory will be the permissions as specified after the **=** operator for the **who** and the remaining **who** retain values as per the **umask**.
- If **mkfifo** is invoked with the **-m** option, the **umask** is not referred for setting the mode permissions if the mode given by you is in octal.
- If the mode you provide is in symbolic notation, only the bits given by you are set. The rest of the bits (for which **who** is not provided), are set as per the value in **umask**.

## **mm(1)**

**mm** uses uncompiled macro files only; compiled macro files are no longer supported. The **-y** option, which specifies uncompiled macro files on 9.x, was not needed as of 10.0.

## **more(1)**

Many new options, and some changed, to comply with XPG4 standards.

Changes include:

- Backward scrolling.
- Third line of screen is current position for positioning commands.

- `-e` causes exit after last line of last file; automatic exit after last line only if `stdout` is not a terminal device.
- `-f` and `-n` imply `-c`.
- Special handling for `^L` (form feed) eliminated.
- `-l` option eliminated; this behavior is now default.

See the manpage for details.

## **mt(1)**

The default device file for this command in 9.x and earlier releases was `/dev/rmt/0mn`. As of 10.0, the default is `/dev/rmt/0mnb`.

The reason for this is that as of 10.0, `/dev/rmt/0mn` represents AT&T mode, whereas only raw, no-rewind Berkeley-style devices should be used with `mt(1)`.

On the Series 700, `/dev/rmt/0mn` used to represent a no-rewind, Berkeley-style device. On 10.x such a device is represented by `/dev/rmt/0mnb`. On the Series 800, `/dev/rmt/0mnb` has always represented this type of device and continues to do so.

System	Pre-10.0	10.x
Series 700	<code>/dev/rmt/0mn</code>	<code>/dev/rmt/0mnb</code>
Series 800	<code>/dev/rmt/0mnb</code>	<code>/dev/rmt/0mnb</code>

If you have 9.x scripts that use `/dev/rmt/0mn` with `mt(1)`, you will need to change them to use `/dev/rmt/0mnb` before running them on a 10.x system.

## **mv(1)**

`mv` will provide warning messages whenever extent attributes of a file are lost. You can use the `-e` option to control what `mv` does if it cannot find extent attributes.

See the entry for “`cp(1)`” in this section for more information. See also “`cpio(1)`”, “`ftio(1)`”, and “`ls(1)`” in this section.



## **nice(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- New option **n** with a “numeric option argument”.
- Changes to usage and error messages.

## **nl(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The options can now be intermingled with the optional file operand.
- If **-** is specified as a file name, input will be taken from **stdin**.
- The limit for input lines is increased to 2048 bytes.
- If text numbering option is used, the text line will now be numbered only if that line contains graphic characters.
- Only one file can now be specified. If multiple files are specified, an error occurs.

## **nljust(1)**

New syntax:

```
nljust [-acilnt] [-d digits] [-e seq] [-j just] [-m mode]
      [-o order] [-r margin] [-w width] [-x ck] [file...]
```

New options:

- i** Triggers ISO 8859/6 interpretation of the data.
- d *digits*** Where *digits* is **h**, **w**, or **b**. Processes digits for output as Hindi (**h**), Western (**w**), or both (**b**).

These options support the ISO 8859/6 character set (also known as **ASMO 708**).

## **nohup(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- If a file is created, the file's permissions bits will be set to `S_IRUSR` or `S_IWUSR`.
- The following exit values are returned:

126                   The command specified by `command` was found, but could not be invoked.

127                   An error occurred in the `nohup` utility or the specified command could not be found.

Otherwise, the exit status of `nohup` will be that of the command specified.

- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **nroff(1)**

New option:

-P                   directs Asian printers to print two-column-wide characters in boxes 1 1/2 columns wide.

## **od(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The output format has been changed. When multiple types are specified using multiple `-bcbox` options, output lines are written for each type in the order specified.
- If multiple files are specified on the command line, the dump is uniformly generated reading the files as if there was one file that was generated by concatenating each file on the command line.

## **pack(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- No packing occurs if the file name has more than `NAME_MAX - 2` characters.

## **patch(1)**

*For 10.10:*

- Reads a patch file containing a difference listing produced by the `diff(1)` command.
- Applies the diff file to an original file, producing a patched file.

See the `patch(1)` manpage for details.

## **pathchk(1)**

*For 10.10:*

- When the length of a given pathname exceeds the system limit, the command will exit with non-zero value.

## **passwd (1)**

New options:

- |                        |   |
|------------------------|---|
| <code>-m (min)</code>  | Sets the minimum time between password changes for a given user.                        |
| <code>-x (max)</code>  | Sets the maximum time before the password expires.                                      |
| <code>-w (warn)</code> | Sets the time prior to expiration when users receive warnings to change their password. |
| <code>-f</code>        | Forces the user to change the password at the next login attempt.                       |
| <code>-F</code>        | Same as the old <code>-f</code> .   |

The above new password aging options will not work with `-F`, and they affect only the system's local users.

## **paste (1)**

No longer limits output line to 1023 characters.

## **pax(1)**

*For 10.20:*

Because of industry standards and interoperability goals, files larger than 2GB cannot be archived using `pax(1)`. You will receive an error message if archival is attempted for a file that is larger than 2GB in size.

User and group IDs larger than 60K are not supported. Files with user/group IDs greater than 60K are archived and restored under the user/group IDs of the current process.

The `fbackup` and `frecover` command set is recommended for backing up files that are larger than 2GB or are owned by user/group IDs that are greater than or equal to 60000.

*For 10.10:*

- New option `-o` provides information to the implementation to modify the algorithm for extracting or writing files that is specific to the file format specified by `-x`.
- `MAXLINE` has been changed from 1024 to 2048.
- `pax` has been changed to restore the file with its original permissions when the file was backed up.

*For 10.0:*

- Based on OSF 1.2.
- Supports 4-byte EUC
- POSIX.2 compliant

New options:

`-k` Do not write over existing files.

**-X** When traversing the file hierarchy specified by a pathname, do not descend into directories that have a different device ID.

Changed options:

Old option	Replaced by	Meaning
-m	-pm	File modification times preserved
-o	-po	File ownership restored from archive
-p	-t	Access times preserved
-t [device]	-f [archive]	Overrides standard default input for -r and standard output for -w.

Option no longer supported:

**-H** will now result in the general usage message. See “NFS Diskless” in Chapter 4 for details.

## **pr(1)**

*For 10.10:*

- A new option **-c** has been added. This option is equivalent to the existing **-k** option.
- With the **-k** option, if an invalid option is used, **pr** does not report an error message.
- **pr** now returns a non-zero exit value for missing option arguments.
- If multiple files are specified with **pr** and the **-F** option, the “\n” character will be added at the end of output.

## **prs(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- The **-c** and **-d** option arguments are now mandatory.

- When “-” is specified as an argument, SCCS directory names given on standard input are processed (they were ignored on earlier releases).
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- If an error occurs when processing one of the files, the processing continues with the next file operand, but exits with a non-zero status.
- Unreadable files are now silently ignored.

## ps(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- New options have been added that increase the data available from **ps** and the ability to customize its’ output. The options that changed are: **-A**, **-c**, **-C**, **-G**, **-H**, **-j**, **-n**, **-o**, **-s**, and **-U**. Refer to the manpages for option details.

Specifically, the changes for XPG4 compliance are:

- Addition of the **-A** option. This option is the same as the existing **-e** option.
- Addition of the **-G** *grouplist* option. This option writes the information for processes whose real group ID numbers are given in **grouplist**.
- Addition of the **-o** *format* option. This option writes information according to the format specification given in **format**.
- Addition of the **-U** *userlist* option. This option writes information for processes whose real user ID numbers or login names are given in **userlist**.
- Addition of the **-n** *namelist* option, which is ignored.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

*For 10.0:*

- Two new options for HP Process Resource Manager (HP PRM)
- Modifications to accommodate DTC nailed ports and Streams **ptys**

New options (for systems configured with HP PRM):

- P                    Adds an additional column to **ps**’s output, showing the process resource group ID of each process.

- R Filters the processes displayed, showing only those that belong to specified process resource groups.

See also “HP Process Resource Manager (HP PRM)” in Chapter 5.

Modifications for DTC nailed ports and Streams `ptys`:

The manner in which `ps` searches for terminal device files has been changed to accommodate DTC nailed ports and Streams `ptys`, which are supported with these restrictions:

- All DTC nailed port device files must appear in the directory `/dev/telnet`.
- All Streams `ptys` must be specified, and will be printed, as `pts/NN`, where `NN` is the `pty` number,

### **renice(1)**

*For 10.10:*

- Arguments are evaluated more strictly.
- Error messages have been added:
  - One error message asks you to enter an unsigned decimal process/group ID whenever you enter an invalid process/group ID.
  - One error message informs you if the process/group ID entered is out of range.

### **rm(1)**

- Changed to comply with XPG4.
- Will descend to arbitrary depth in file hierarchy without path length limitation.
- Now conforms to XPG4 specifications:

Changed options:

Option	Changed Behavior
-f	Any previous occurrence of the -i option will be ignored.
-i	Any previous occurrence of the -f option will be ignored.
-r, -R	(Minor changes on encountering directories without read, write or execute permission. See the manpage for details.)

## **rmde1(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- Silently ignores unreadable files.
- When “-” is specified as an argument, SCCS directory names given on standard input are processed (they were ignored on earlier releases).

## **rmdir(1)**

Changed options:

Option	Changed Behavior
-f	Any previous occurrence of -i option will be ignored.
-i	Any previous occurrence of -f option will be ignored.



## **sact(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- When “-” is specified as an argument, SCCS directory names given on standard input are processed (they were ignored on earlier releases).
- Unreadable files are silently ignored.
- Input buffers of size `BUFSIZ` have changed to `LINE_MAX`.
- When more than one file (operand) is given on the command line and an error occurs in one of the files, the utility continues processing the other file and exits with a non-zero exit status.

## **sar(1)**

`sar` has a new option, `-S`, to report `select` calls.

`sar -m` reports message and semaphore activities. This report includes the number of System V `msgrcv()` calls per second and the number of System V `semop()` calls per second. By explicitly specifying the new `-S` option with `-m`, you can make `sar` report the number of System V `select()` calls per second as well.

The default behavior of `sar -m` has not changed.

## **sccs(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The relative path name can be assigned to the `PROJECTDIR` environment variable.

*For 10.01:*

Front end to the programs comprising the Source Code Control System. Provides short cuts to running the SCCS commands directly.

## sed(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- Error messages are redirected to `stderr`.
- `sed(1)` handles eight-character labels.
- A call to `sed -f script_file` accepts a script file consisting of editing commands, one per line.
- When the editing command `D` deletes the whole current pattern space, the next cycle of editing commands is applied to the remaining pattern space.
- The editing command `H` appends to the hold space a *newline* followed by the contents of the pattern space.
- `r pathname` returns with zero exit status for correct operation.
- `w pathname` returns with zero exit status for correct operation.
- The editing command `x` switches the contents of the pattern and hold spaces.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

*For 10.0:*

- General usage message displayed in response to invalid options.
- Improvements to `l` output.
- Invalid regular expressions not processed (for example the expression `[x[=]` will not be processed as of 10.0).
- Only printable multi-byte characters printed; non-printable characters displayed in octal.

## sh-posix(1)

*For 10.10:*

- Two new signals, `SIGXCPU` and `SIGXFSZ` are available in POSIX `sh` (`/usr/bin/sh` and `/sbin/sh`).
- The history file has a new internal magic number that causes a pre-10.10 history file to be discarded. If you use multiple versions of HP-UX that NFS-mount the same directory containing the history file (usually the home directory), you may want to have a separate history file for each HP-UX version by setting the proper `HISTFILE` parameter. Read the `sh-posix(1)` manpage `WARNING` section for more details.

- The file descriptors used internally by the POSIX shell are moved to numbers 24 through 30.
- The `cd` built-in command—if the new current directory is found using a non-null pathname from shell parameter `CDPATH`, an absolute path of the new current directory is written to standard output, regardless of whether the shell is executing a script or non-interactive. In previous releases, the path was written only by interactive shells.
- The `fc -l` built-in command—when using the `-l` option of the `fc` built-in command, if the argument `last` is omitted, the last command listed is the previous command. In previous releases, the last command listed was the `fc -l` command currently being executed.
- The `whence -v` command—if a `whence -v name` (or “type name”) generates the error “name not found”, the message will be displayed to `stderr`. In previous releases, the message was displayed to `stdout`.
- The `getopts` built-in command—whenever the shell is invoked, `OPTIND` will be initialized to 1.
- When an I/O redirection error occurs during a call to a function, a non-interactive shell aborts. In previous releases, only the currently executing function was aborted.
- The POSIX `sh` supports the POSIX internationalization module. You have to set the proper `LC_*` or `LANG` environment parameters to input and output the proper local languages.

Refer to the `sh-posix(1)` manpage for details.

## **sort(1)**

New option:

- A                 Sorts on a byte-by-byte basis using each character’s encoded value. Extended (signed byte) characters will be considered negative values, and sort before ASCII characters. If you are sorting ASCII characters in a non-C/POSIX locale, this flag performs much faster.

Changed options:

Option	Changed Behavior
=====	
-f	Supports folding of multibyte characters, if folding is possible in the locale.
-t	Supports multibyte characters as field delimiters.

Option no longer supported:

-l (previously ignored).

### **split(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- `split -l` returns with status zero when all files have been successfully created and an EOF has occurred in the input file.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

### **strings(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- `strings(1)` now exits with a non-zero status on failure.

### **strip(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Now returns non-zero status when an error occurs, but continues processing the remaining files.

## **sum(1)**

Change to `-p` option.

The checksum values provided when you specify `-p` are generated according to a new algorithm to ensure conformance with the latest POSIX.2 standard.

This means that checksum values will differ between 9.x and 10.x for the same input. If you need to obtain checksum values calculated by the 9.x algorithm, you can use the old version of the command, delivered with 10.x as `/usr/old/usr/bin/sum`. If you are using these values in a script, use `/usr/old/usr/bin/sum` to verify them, then update them to the 10.x values using the 10.x command.

`-p` now also supports piped input.

## **tar(1)**

*For 10.20:*

Because of industry standards and interoperability goals, files larger than 2GB cannot be archived using `tar(1)`. You will receive an error message if archival is attempted for a file that is larger than 2GB in size.

User and group IDs larger than 60K are not supported. Files with user/group IDs greater than 60K are archived and restored under the user/group IDs of the current process.

The `fbackup` and `frecover` command set is recommended for backing up files that are larger than 2GB or are owned by user/group IDs that are greater than or equal to 60000.

*For 10.10:*

- New `e` option
- Can extract multiple archives from one tape using a no-rewind device

New option:

The new option `e` causes `tar` to fail whenever extent attributes of a file are lost.

See also “`cp(1)`”, “`cpio(1)`”, “`ftio(1)`”, “`ls(1)`” and “`mv(1)`”.

## Extracting Multiple Archives

As of 10.0, `tar` can extract multiple archives from one tape when using no-rewind device files. This requires different steps depending on whether you are using an AT&T-style or Berkeley-style device file.

- Using AT&T-style device file with no-rewind:

```
mt -t /dev/rmt/Omn rew      rewind to beginning of tape
tar -xvf /dev/rmt/Omn      extract first archive
tar -xvf /dev/rmt/Omn      extract second archive
tar -xvf /dev/rmt/Omn      extract third archive
```

- Using a Berkeley-style device file with no-rewind:

```
mt -t /dev/rmt/Omnb rew     rewind to beginning of tape
tar -xvf /dev/rmt/Omnb      extract first archive
mt -t /dev/rmt/Omnb fsf     position to beginning of second
                             archive
tar -xvf /dev/rmt/Omnb      extract second archive
mt -t /dev/rmt/Omnb fsf     position to beginning of third
                             archive
tar -xvf /dev/rmt/Omnb      extract third archive
```

- Using a Berkeley-style device file with no-rewind to restore the first and third `tar` archive:

```
mt -t /dev/rmt/Omnb rew     rewind to beginning of tape
tar -xvf /dev/rmt/Omnb      extract first archive
mt -t /dev/rmt/Omnb fsf 2   position to beginning of third
                             archive
tar -xvf /dev/rmt/Omnb      extract third archive
```

## **time(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Addition of the option `-p`, which provides a usage message for `time(1)`.
- Context-sensitive error message have been added to reflect the cause of errors in `time(1)`.

- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- Exit values of `time(1)` have changed. If the command to be timed was found, but could not be invoked, the exit value is 126. If the command to be timed could not be found, the exit value is 127.
- Exit values of `time(1)` when a signal occurred has been corrected.

## **top(1)**

*For 10.10:*

- Processes whose size exceeds 99999Kb will have their size reported in megabytes, with an `M` suffix.
- Scripts and programs that parse top output will have to be sensitive to the units in use to properly handle large processes.

## **tput(1)**

*For 10.10:*

- The maximum length for the input line has changed from `BUZSIZ` to `LINE_MAX`.

*For 10.0:*

- SVR4-compatible `-S` option.
- Improved performance.
- Supports many new terminal capabilities.

New option:

`-S`                    Allows terminal capabilities to be read from `stdin`.

See the manpage for details.

## tr(1)

- New, XPG4-compliant syntax
- New -A option

New syntax:

```
tr [-Acs] string1 string2
tr -s [-Ac] string1
tr -d [-Ac] string1
tr -ds [-Ac] string1 string2
```

New option:

- A Translates on a byte-by-byte basis. `tr` with this option does not support extended characters.

Changed options:

- Some changes affecting `-c`, `-d`, and `-s`.

If `-c` and `-d` are both specified, all characters except those specified by *string1* are deleted. The contents of *string2* are ignored, unless `-s` is also specified.

The same string cannot be used for both the `-d` and the `-s` flags.

If `-d` is not specified, each input character or collating element found in the array specified by *string1* is replaced by the character or collating element in the same relative position in the array specified by *string2*.

The abbreviation `[.cc.]` is no longer supported because multi-character collating elements no longer require special treatment.

- Changes affecting the following abbreviations used to introduce ranges of characters or repeated characters into the strings:

- `[:class:]` or `[[:class:]]`

Represents all characters belonging to the defined character class, as defined by the current setting of the `LC_CTYPE` locale category.

- `[=equiv=]`

Can be used for *string1* or *string2* only in combination with the `-d` and `-s` flags.



- [a\*n]

Valid only when it occurs in *string2*.

## **tsort(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **uncompress(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Now returns a value of 1 whenever **uncompress** fails on one of the operands. (In previous releases, a return status of 0 was returned.)
- Now if the **uncompress** command without the **-f** option executes on an existing file and the file is not altered, **uncompress** exits with a status of 1. (In previous releases, a return status of 0 was returned.)
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

## **unexpand(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- When an error occurs processing one of the operands, processing continues with the remaining operands and the final exit status is non-zero.

## **uniq(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Can now handle a maximum of 2048 characters per line (includes **newline**).

*For 10.0:*

The `-s` option assumes each line in the input is terminated with a new line character for purposes of comparison.

## **unpack(1)**

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- The maximum file name length changed to take `NAME_MAX - 2` as the maximum filename length.
- The maximum path length has changed to `MAXPATHLENGTH`.

## **uucp(1)**

*For 10.01:*

As a consequence of the new NFS file-sharing paradigm, the `uucp(1)` and `cu(1)` commands are now supported on client systems.

This change applies to all commands and scripts in the UUCP fileset, as well as to the commands in the SYSCOM fileset:

The files in the UUCP fileset are as follows:

```
uucp
uux
uulog
uupick
uustat
uuto
uuls
uusnap
uusnaps
mkuupath
uucpd
getx25
SetUp
Uutry
```

```
uucheck
uucico
uuclean
uucleanup
uugetty
uuname
uusched
uusub
uuxqt
dialit
remote.unknown
uudemon.admin
uudemon.cleanu
uudemon.hour
uudemon.poll
opx25
clrsvc
```

The files in the SYSCOM fileset are as follows:

```
cu
ct
uupath
kermit
umodem
vt
vtdaemon
vtgateway
vtserver
```

In a 9.x “DUX” cluster, all systems have a shared root and sharing directories. `ct` and `cu` can be run only from a root server and therefore can utilize only root server physical ports. If you attempt to run `cu` on a 9.x cluster client, this message is returned: **cu must be run on the root server**. UUCP physical connections are restricted to the root server. UUCP can be run from a client, but is actually using root server UUCP configuration, physical ports, and device files.

In 10.01 clusters, all systems have a private root directory, and all other directories are private by default. All systems (standalone, server, clients)

are treated identically. UUCP, `ct`, and `cu` are supported on any system (standalone, server, or client). There is no distinction between the different system configurations—standalone, server, diskless client, or client with local disk. Consequently, each system has his own private configuration, log files, lock files, spool and work files. The only shared files between server and clients, are the command executables, backend executables, system administration commands, and startup/shutdown scripts. When you execute a command, the command is accessing the physical ports and associated device files on the system where the command was issued (for example, if you run UUCP from a client, you are using the client device files, physical ports and UUCP configuration).

---

**Note** To run UUCP from an NFS client and use the server device files, physical ports, and UUCP configuration (that is, produce 9.0 behavior), `telnet` to the server, and run UUCP.

---

## **uuencode(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **uulog(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- If a system is not specified as the command argument with the `-s` option, the exit code is 2. In previous releases, the return value was 0.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

## **uupick(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- **uupick(1)** now takes care of invalid parameters such as:
  - when the argument to **-s** is **NULL**. That is, the system name is not specified.
  - when the number of parameters exceeds two.
- The error messages are now redirected to **stderr**.
- In case of errors, the utility exits with a non-zero exit value.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

## **uustat(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- If no system is specified as the command argument with the **-s** option, error messages are written to **stderr** with a return value of 1. In previous releases, no error messages were written and the exit status was 0.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

## **uuto(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The exit code changed to a non-zero value if there is an invalid filename argument.
- The error message is printed to **stderr** instead of **stdout**.
- “**--**” as an argument indicates the end of options and the beginning of operands. Any arguments following the “**--**” that starts with a “**-**” are considered to be operands (not options).
- If a FIFO special file is specified as an argument, the command exits with an exit code 2. In previous releases, this was ignored.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

## **uux(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- Addition of the `-p` option.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

## **val(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- The limit on the number of files that can be specified on the command line has been removed. In previous releases, the number was limited to 50.
- “--” as an argument indicates the end of options and the beginning of operands. Any arguments following the “--” that starts with a “-” are considered to be operands (not options).

## **vi(1)**

*For 10.20:*

`vi(1)` is XPG4 compatible. This compliance is achieved by incorporating the changes listed below. These changes will occur only if the `UNIX95` environment variable is set.

Changes:

- The `abbreviation` or `ab` command now allows abbreviating recursively.
  - The size of the `tmp` files created when `vi`, with the `LANG` environment variable set to any language other than C, was previously very large. At 10.20, this is no longer the case.
- \* The XPG4 changes include the following :
- If the implementation supports the `ctags` utility and when `-t tagstring` is specified with either the `-c` command or `+ command`, `vi` processes `-t tagstring` first and then executes the command.

- When P or p is entered in the command mode and the unnamed buffer does not contain complete lines, vi inserts text from the unnamed buffer before the current cursor position and positions the cursor to the last character of the inserted text.
- When x is a lowercase character in the POSIX locale and the buffer referenced by x contains lines from a previous put or yank command, line put places the lines in the buffer referred to by x after line (the default is the current line) and sets the current line indicator to the first of the inserted lines.
- All error and warning messages should be written to standard error.
- When the warn option is set, a ! command is issued, and the contents of the current buffer is modified, vi writes a warning to standard error before executing the ! command.
- When W is entered in command mode and there is no following bigword on the current line, vi moves to the first character of the first bigword on the next line that contains a bigword.
- When count e is entered in command mode and the cursor position is at the end of a word, vi moves the cursor to the last character of the count-th following word (the default is the next word).
- When U or u is entered in command mode and the autoindent option is not set, vi restores the line to its state before the cursor was last moved and positions the cursor to column position one.
- When count { is entered in command mode and there are count paragraphs before the current cursor position, vi moves backward to the beginning of the count-th preceding paragraph, as defined by the ex paragraph option. The current line is set to the line containing the beginning of that paragraph. The current column is set to the first nonblank character of that paragraph.
- When control-D is entered in command mode and the line movement would place the current line after the end of the file, the movement does not occur. The terminal is alerted and the current position is not changed.

### Impact

The above changes impact the backward compatibility of the vi(1) command. These changes, however, will come into effect only when the UNIX95 environment variable is set.

## Compatibility

The changes made will render this command binary incompatible. The incompatible behavior will be seen only when you set the `UNIX95` environment variable.

*For 10.10:*

- The addition of the `-c` option, which is equivalent to the `+` command.
- Now reports an error when the `-s` option is used.

## `wc(1)`

*For 10.10:*

- Now conforms to XPG4 specifications.
- Now accepts vertical tab (`\v`), form feed (`\f`), and carriage return (`\r`) as delimiters for words.
- The output of `wc` is in this format:  
*(Number of lines) (Number of words) (Number of characters) (file name)*

This format is independent of:

- The order in which the options are specified. Previously, `wc` printed in the same order as the options appeared in the command line.
- The number of repeated options specified. Previously, `wc` printed as many line, word, character, byte counts as the number of options appeared in the command line.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

## `what(1)`

*For 10.10:*

- Now conforms to XPG4 XBD Utility Syntax Guidelines.
- `--` as an argument indicates the end of options and the beginning of operands. Any arguments following the `--` that starts with a `-` are considered to be operands (not options).
- Returns an error for invalid options.



## who(1)

*For 10.10:*

- Now conforms to XPG4 specifications.
- The XPG4 behavior is activated if you set the UNIX95 environment variable.
- Changes:

Option	Description
-H	Displays the header if used with the <code>-d</code> , <code>-r</code> and <code>-b</code> options. Displays the header before data if used with the <code>-a</code> option.
-T	Displays the state of the header if used with the <code>-T</code> option. Will not display information about the <code>IDLE</code> , <code>PID</code> and <code>COMMENTS</code> fields. Will not display information about <code>PID</code> and <code>COMMENTS</code> if used with the <code>-u</code> option.
-s	When used with <code>-u</code> options, displays the terminal state. Is mutually exclusive with the <code>-T</code> , <code>-a</code> and <code>-d</code> options.
-q	If used, other options will be ignored.
who am i	Displays usage if called with option(s).
who am I	Displays usage if called with option(s).

## xargs(1)

*For 10.10:*

- Addition of `-I`, `-E`, and `-L` options.
- The `-i`, `-e`, and `-l` options have optional arguments.
- Exit codes are returned on non-zero failures.
- The command can take input arguments up to 2048 (`LINE_MAX`) characters.
- If the utility specified on the command line was found, but could not be invoked, `xargs` returns an exit code of 126.
- The command returns a value of 127 when the utility could not be found.
- The exit code of a signal to `xargs` is within the range of 0 to 127 or -1. In previous releases, the exit code was greater than 128.
- The XPG4 behavior is activated if you set the UNIX95 environment variable.

## **zcat(1)**

*For 10.10:*

- Now conforms to XPG4 specifications.
- **zcat** returns a value of 1 whenever it fails on one of the operands. In previous releases, it returned a status of 0.
- The XPG4 behavior is activated if you set the **UNIX95** environment variable.

---

## **Changed Commands in Section 1M**

This section summarizes changes to commands in Section 1M of the *HP-UX Reference* that are not discussed elsewhere in this document.

Commands changed at 10.20:

- **fuser(1M)**
- **mount(1M)**
- **ncheck\_hfs(1M)**
- **unmount(1M)**

Commands changed at 10.10:

- **bdf(1M)**
- **df(1M)**
- **frecover(1M)**
- **savecore(1M)**
- **shutdown(1M)**

Commands changed at 10.01:

- **catman(1M)**
- **savecore(1M)**
- **swapon(1M)**

Commands changed at 10.0:

- **config(1M)**
- **dcopy(1M)**
- **dodisk(1M)**
- **fbackup(1M)**

- `ff(1M)`
- `fsck(1M)`
- `fsdb(1M)`
- `fuser(1M)`
- `ioscan(1M)`
- `labelit(1M)`
- `lsdev(1M)`
- `localedef(1M)`
- `mkboot(1M)`
- `mkfs(1M)`
- `mount(1M)`, `umount(1M)`
- `ncheck(1M)`
- `newfs(1M)`
- `quot(1M)`
- `quotacheck(1M)`
- `sysdef(1M)`
- `syslogd(1M)`
- `volcopy(1M)`
- `whodo(1M)`

## File System Administration Commands

The following commands now consist of a “wrapper”, a script that determines the file system type and then executes the code appropriate to that type:

- `ff(1M)`
- `fsck(1M)`
- `fsdb(1M)`
- `labelit(1M)`
- `mkfs(1M)`
- `mount(1M)`
- `ncheck(1M)`
- `newfs(1M)`
- `quot(1M)`
- `volcopy(1M)`

See the entries for these commands later in this section. For details of the “wrapper”, see the `fs_wrapper(5)` manpage on a 10.x system.

You will need to specify the file-system type, either on the command line for individual commands, or in `/etc/fstab`.

(If you do not, the commands will look in an HP-supplied file called `/etc/default/fs`; this may produce the result you want, but you should not rely on it.)

The file system-specific commands are stored in `/sbin/fs/[FStype]` or `/usr/sbin/fs/[FStype]`. They are not meant to be executed directly.

## **bdf(1M)**

*For 10.10:*

- `bdf(1)` will not report negative values for the field `avail` when the disk gets full. The value will always be greater than or equal to zero.
- The `%used` field will not report values greater than 100. The value will always be less than or equal to 100.

*For 10.01:*

- Supports multiple file systems
- Output format changed
- `-t`, `-i` options changed
- `-L` no longer supported

Changed options:

Option	Changed Behavior
<code>-t [type]</code>	Extended to work with all supported file system types.
<code>-i</code>	Displays <code>-1</code> as number of free inodes for file systems that do not have a fixed number of inodes.

Option no longer supported:

9.x Option	10.x Replacement	Comments
=====		
-L	(none)	HP-UX "DUX" clusters were not supported as of 10.0.

Output format change:

The header now aligns with the statistics, and there is always a space between fields.

## catman(1M)

`catman` has a new option at 10.01, `-A <alt-root>`, to support client/server diskless environments. Under 10.01, diskless clients mount `/usr` read-only, and they cannot make changes to `/usr/share/man`. Therefore, `catman(1M)` can only be run on a server, and you may need to run `catman` with the `-A` option, depending on where the clients' manpages are located:

- If the manpages are being served to a client, the client's manpages are located on the server in (for example):

```
/export/shared_roots/OS_700/usr/shared/man/
```

If the client and server are both 10.01, Series 700 machines, then the link `/export/shared_roots/OS_700 -> /` is present and the client and server have common manpages. In this case, simply run `catman(1M)` on the server.

- If the client does not share manpages with the server, use the new `-A <alt-root>` option to produce `catman` pages for a client. For example:

```
catman -A /export/shared_roots/OS_700
```

In this example, `cat*` manpages would be created in these (default manpaths):

```
/export/shared_roots/OS_700/usr/share/man/cat*  
/export/shared_roots/OS_700/usr/local/man/cat*  
/export/shared_roots/OS_700/usr/contrib/man/cat*
```

The `whatis` database would be created in `/export/shared_roots/OS_700/usr/share/lib/whatis`.

## **config(1M)**

*For 10.0:*

- Supports kernel build on Series 800 (replacing `uxgen`) as well as 700
- Supports I/O convergence and new file system layout
- Uses `system` file as input (similar to Series 700 `dfile`)
- `-a` option no longer supported
- New `-s` option
- Only one swap device can be specified

`config` is the basis for all kernel builds on 10.x systems.

---

<b>Note</b>	HP recommends you use <code>mk_kernel(1M)</code> or SAM (the menu-driven System Administration Manager) rather than <code>config</code> itself, to rebuild and install the kernel.
-------------	--

---

New option: `-s`

`-s` prevents `config` from calling `make` to build the kernel; `conf.c` and `conf.mk` are output.

Option no longer supported: `-a`

This option produced a script to help make device files. `insf(1M)` and `mksf(1M)` (supported on both the Series 700 and S800 platforms) are the best way to make device files on a 10.x system.

### **Input and Output Files**

`config` accepts a `system` file as input, usually `/stand/system`. This file is similar in format to the 9.x Series 700 `dfile`, but the `root` and `console` device statements are no longer used. By default `config` writes its output files into the directory `/stand/build` (if the input file is `/stand/system`) and calls `make(1)` to compile and link a kernel, `/stand/build/vmunix_test`.

The `/etc/master` file was not supported as of 10.0. By default `config` now searches the directory `/usr/conf/master.d` and reads all the files in it for device specifications, tunable specifications, etc. See the `master(4)` manpage for a description of the master files.

## dcopy(1M)

---

**Note** Release 10.10 will be the last release that `dcopy(1M)` will be supported.

---

*For 10.0:*

- Complies with SVR4
- Available on both Series 700 and 800

New options:

- F [**FSType**] Specifies which file system type the command is to operate on. For 10.0, `dcopy` supports only HFS.
- v Verbose. Echoes the completed command line.

As of 10.0, this command is stored in `/usr/sbin/dcopy`.

## df(1M)

*For 10.10:*

- Now conforms to XPG4 specifications.
- Addition of the `-P` option. This option produces output that indicates the following:

<i>file system name</i>	Name of the system
<i>total space</i>	Size of the file system
<i>space used</i>	Total amount of space allocated to existing files in the file system
<i>space free</i>	Total amount of space available within the file system for creating new files by unprivileged users
<i>percentage used</i>	Percentage of the normally available space that is currently allocated to all files on the file system
<i>file system root</i>	Directory below which the file system hierarchy appears.
<i>lf</i>	New line character

- The `-P` option is mutually exclusive with the `-t` option.



- `df(1M)` does not report negative values for the field `avail` when the disk gets full. The value will always be greater than or equal to 0.
- The `%used` field will not report values greater than 100 (previous releases did).

*For 10.0:*

- Complies with SVR4
- Supports multiple file systems

`df` now has both generic and file system-specific syntax and options. Many of the options were new for 10.0, including `-F[FSType]`, `-V` (verbose), `-e` (files free), `-g` (`statvfs` report) `-i` (inodes report), `-k` (allocation in KBytes) `-n` (file system name), `-o` (file system-specific options), `-v` (blocks report) . For details, see the manpages or the *HP-UX Reference*, under `df` for generic usage and `df_[FSType]`, for file system-specific usage—for example, look under `df_hfs(1M)` for HFS behavior.

Changed option:

	Old	New	New Option
Option	Behavior	Behavior	To Get Old Behavior
<code>-b</code>	Report total number of blocks for file system swap, allocated and free	Report number of KBytes free	<code>-B</code>

You will need to specify the type of the file system `df` is to operate on—either on the command line or in `/etc/fstab`.

As of 10.0, this command is stored in `/usr/bin/df`. The 9.x version of the command is still available under `/usr/old`.

### **dodisk(1M)**

This command has been modified to support multiple file systems. The changes will not affect existing (9.x) scripts.

## **fbbackup(1M)**

*For 10.0:*

- Supports files that have extent attributes.
- Supports loopback file systems (LOFS)

New options:

- E Ignore extent attributes. (The default is to save them.)
- l Back up LOFS files. (The default is to not to back them up.)

**frecover(1M)** will recover extent attributes. Other commands that support archiving extent attributes include **cpio(1)** and **ftio(1)**. See the entries for those commands in this chapter, and the manpages.

## **ff(1M)**

*For 10.0:*

- Complies with SVR4.
- Supports multiple file systems.
- Available on both Series 700 and 800.

**ff** now has both generic and file system-specific syntax and options. For details, see the manpages or the *HP-UX Reference*, under **ff** for generic usage and **ff\_[FSType]** for file system-specific usage—for example, look under **ff\_hfs(1M)** for HFS behavior.

New options:

- F [FSType] Specifies which file system type the command is to operate on.
- o [specific\_options] Specifies file system-specific options, if any.
- V Verbose. Echoes the completed command line.

You will need to specify the type of the file system **ff** is to operate on—either on the commandline or in */etc/fstab*. 10.x **ff** consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the **fs\_wrapper(5)** manpage for more information.

As of 10.0, this command is stored in `/usr/sbin/ff`.

## **frecover(1M)**

*For 10.10:*

The `frecover(1M)` command might experience intermittent name cache failure. To offset this intermittent failure, the directory/file name is added to standard error even if this condition is encountered.

*For 10.01:*

Changed to support recovery of files that have extent attributes.

New option:

`-E [extarg]` Specifies how to handle extent attributes and associated errors and warnings. (The default is to recover extent attributes.)

## **fsck(1M)**

*For 10.0:*

- Complies with SVID3
- Supports multiple file systems

`fsck` now has both generic and file system-specific syntax and options. For details, see the manpages or the *HP-UX Reference*, under `fsck` for generic usage and `fsck_[FSType]` for file system-specific usage—for example, look under `fsck_hfs(1M)` for HFS behavior.

New options:

- F [FSType] Specifies which file system type the command is to operate on.
- V Verbose. Echoes the completed command line.
- m Causes **fsck** to return zero if the file system is mountable, 32 if it needs additional checking, 33 if it is mounted, or a higher number if it is badly damaged.
- c 0 Eliminates use of the cache, reducing **fsck**'s memory needs and allowing it to run on systems with 16Mb of memory. Not recommended for systems with more than 16Mb of memory because of the performance degradation.
- o [specific\_options] Specifies file system-specific options, if any.

Changed option:

	Old	New	New Option
Option	Behavior	Behavior	To Get Old Behavior
	=====		
-F	Check mounted file system	Specify file system type	-f

You will need to specify the type of the file system **fsck** is to operate on—either on the command line or in `/etc/fstab`. 10.x **fsck** consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

As of 10.0, this command is stored in `/usr/sbin/fsck`.

## 7-70 Commands and Libraries

## **fsdb(1M)**

*For 10.0:*

- Complies with SVID3
- Supports multiple file systems

`fsdb` now has both generic and file system-specific syntax and options. For details, see the *HP-UX Reference*, under `fsdb` for generic usage and `fsdb_[FSType]` for file system-specific usage—for example, look under `fsdb_hfs(1M)` for HFS behavior.

New options:

- |                                    |  |
|------------------------------------|--|
| <code>-F [FSType]</code>           | Specifies which file system type the command is to operate on. |
| <code>-V</code>                    | Verbose. Echoes the completed command line.                    |
| <code>-o [specific_options]</code> | Specifies file system-specific options, if any.                |

You will need to specify the type of the file system `fsdb` is to operate on—either on the commandline or in `/etc/fstab`. 10.x `fsdb` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

## **fuser(1M)**

*For 10.20:*

**fuser(1M)** reports the processes that has memory-mapped the files that are passed as arguments to the process. The letter “m” will denote the memory-mapped processes.

**fuser(1M)** output format will be different if a file given as the argument to a process has been memory-mapped by any process. Programs or scripts that are dependent on the format will have to take into consideration the new letter “m” in the output.

*For 10.0:*

- Complies with SVID3
- Two new options
- Report output changed

New options:

- c                Specifies that the directory is a mount point; any usage in that file system is reported.
- f                Causes target to be treated as a regular file, even if it is a file system.

Report output changes:

- t                Indicates that this file is being used by a process as a text file.
- o                Indicates open file.

These changes will affect scripts that explicitly direct **stderr** and **stdout** to the same file.

## **insf(1M), lssf(1M), mkssf(1M), rmsf(1M)**

*For 10.0:*

- Reflect Series 700/800 converged IO; support HP-UX core drivers on 700
- Look for device files in **/dev** by default
- **-l [lu]** replaced by **-I [instance]**
- New options **-D, -q, -v**

New options:

- D *directory*                      Specifies the directory where the device files are to be found. Default is */dev*.
- q                                      Quiet. Suppresses output.
- v                                      Verbose. Increases output.

Options no longer supported:

9.x Option	10.x Replacement	Comments
=====		
-l [lu]	I [instance]	Logical Unit (LU) numbers no longer supported.
-N [cnode]	(none)	HP-UX "DUX" clusters were not supported as of 10.0.

Series 700 and 800 Convergence:

As of 10.0, these commands were supported in the same way on the Series 700 and 800, the only difference being which drivers are available. *insf* creates a device file for the “whole disk”, not for each section or hard partition.

For more information, see “I/O Convergence” in Chapter 4 and the *Configuring HP-UX for Peripherals* manual.

---

**Note**                      *rmsf* will not remove most core drivers from the system, and will output warnings to that effect (unless you run it with *-q*).

---

## **ioscan(1M)**

*For 10.0:*

- Reflects Series 700/800 converged IO
- -l [lu] replaced by -I [instance]
- New option -F

As of 10.0, *ioscan* behaves the same way on the Series 700 and 800; the 9.x Series 800 command is the basis.

New options:

**-F** Lists additional information which can be used to create customized listings via `awk(1)`.

Changed option:

**-f** Behaves the same on Series 700 and 800. Provides more information, in more readable form, than at 9.x.

Option no longer supported:

9.x Option	10.x Replacement	Comments
=====		
<code>-l [lu]</code>	<code>I [instance]</code>	Logical Unit (LU) numbers no longer supported.

For more information, see “I/O Convergence” in Chapter 4 and the *Configuring HP-UX for Peripherals* manual.

## **labelit(1M)**

*For 10.0:*

- Supports multiple file systems
- New options `-F`, `-V`

New options:

**-F [FSType]** Specifies which file-system type the command is to operate on. Required for tape.

**-V** Verbose. Echoes the completed command line.

Except in the case of a tape, you can omit `-F` on the command line if you have defined the file-system type in `/etc/fstab`. 10.x `labelit` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.



## localedef(1M)

*For 10.0:*

`localedef` has undergone major changes. These are summarized below; see the manpage for details.

- Grammar changes to comply with XPG4 and POSIX
- Several new options (see below)
- `-d` and `-o` options no longer supported
- Locales are shared objects rather than binary files

New options:

- |                                    |  |
|------------------------------------|--|
| <code>-C [compiler_options]</code> | Passes the specified options to the compiler that builds the locale. |
| <code>-L [linker_options]</code>   | Passes the specified options to the linker that builds the locale.   |
| <code>-m [method_file]</code>      | Specifies a method file.   |
| <code>-v</code>                    | Displays additional information for debugging.                       |
| <code>-w</code>                    | Displays duplicate-definition warnings.                              |

Options no longer supported:

- |                 |  |
|-----------------|--|
| <code>-d</code> | No longer needed because locale description source files are provided on the system (under <code>/usr/lib/nls/loc/src</code> ). See comments below on the effects of 10.x locales on 9.x code. |
| <code>-o</code> | Not needed because the <code>-d</code> option is no longer supported.  |

---

### Note

If you run `prepare(update_aid)` on a script that invokes `localedef`, you may get a message that the `-n` option is no longer supported. This is incorrect; `-n` is supported on 10.x. (See the *Upgrading from HP-UX 9.x to 10.x* manual for more information on `prepare`.)

---

Grammar changes:

- Overall:
  - Characters in portable codesets no longer need to be quoted (e.g., A need not be specified as 'A');
  - The recommended approach is to use symbolic names for characters, as defined in the `charmap` files.
- Header section: `version`, `hpversion` no longer supported.
- `LC_ALL` category: `context`, `direction` moved to `LC_CTYPE`.
- `LC_CTYPE` category: `bytes_char`, `code_scheme`, `cswidth` no longer supported.
- `LC_COLLATE` category: `modifier` no longer supported.
- `LC_MONETARY` category: `crncystr` no longer supported.

#### **Effect of 10.x Locale Changes on 9.x Code**

**If 9.x Application Uses Customized 9.x Locales, and Linked Shared.** If you have customized 9.x locales, you will need to modify the corresponding 10.x locales to get the same results on 10.x for applications that are linked shared. HP is providing a special 9.0-based `localedef` under `/usr/old/usr/bin` on 10.x systems for this purpose. This special `localedef` supports only the `-d` option to dump a locale.

Dump your 9.x-based locale, then apply the 9.x customizations to the 10.x locale source. This is safer than using the 9.x locale as the base and applying 10.x changes to it, and should be easier in most cases.

For example, to create a customized 10.x locale which in 9.x was based on the `german` locale, do the following:

1. Dump the customized 9.x locale:

- On a 9.x system:

```
localedef -d customized_locale_name > ldf1
```

- On a 10.x system:

```
/usr/old/usr/bin/localedef -d customized_locale_name > ldf1
```

2. Dump the closest standard 9.x locale (`german` in this case):

- On a 9.x system:

```
localedef -d german > ldf2
```

- On a 10.01 system:

```
/usr/old/usr/bin/localedef -d german > ldf2
```

3. Identify the differences between the two dumps:

```
diff ldf1 ldf2
```

4. Use a text editor to apply the changes identified in the previous step to the 10.x `german` locale description file (`/usr/lib/nls/loc/src/de_DE`).

**If 9.x Application Uses Standard 9.x Locales, and Linked Shared.** Applications that were linked shared on 9.x will use the 10.x locales. End-users may see some differences because of changes to locales.

**If 9.x Application Linked Archived.** If the 9.x application was linked archived on 9.x, it will not be affected by 10.x locale changes, whether or not you customized 9.x locales; 9.x locales are still supported in 10.x, though this support will not necessarily continue beyond 10.01.

See the “Locales” section in the *README for NLS in 10.01* for more information on changes affecting NLS locales.

## **lsdev(1M)**

*For 10.0:*

- Reflects Series 700/800 convergence
- New options to report by driver name, class and major number

`lsdev` without options behaves the same as it did 9.x Series 800 systems.

The major difference between 9.x and 10.x on the Series 700 is that the device driver information is extracted from the running kernel instead of a static table. The `Class` column in the output is also new on the Series 700.

## **mkboot(1M)**

*For 10.0:*

- Reflects Series 700/800 convergence
- All options supported on both Series 700 and Series 800
- Supports three disk layouts on both Series 700 and Series 800: LVM, hard partitions, and whole-disk access.
- Ensures that LVM target always has a label
- Default boot program is the same for Series 700 and Series 800:  
`/usr/lib/uxbootlf`
- By default uses the existing disk layout on the target, or LVM if no existing file system

New options:

- H                Specifies that target has hard partitions.
- W                Specifies that target has whole-disk layout.

Changed options:

- c                Additional checks: `-i` and `-p` options.
- f                Restricted to whole-disk layout; no change otherwise.
- h                Restricted to whole-disk layout; no change otherwise.

Options no longer supported:

- s                Series option; no longer needed.

---

**Note** LVM is the default layout if the target disk has no file system.

---

## mkfs(1M)

*For 10.0:*

- Complies with SVID3
- Supports multiple file systems
- Behaves the same on Series 700 and 800

**mkfs** now has both generic and file system-specific syntax and options. For details, see the *HP-UX Reference*, under **mkfs(1M)** for generic usage and **mkfs\_[FSType]** for file system-specific usage—for example, look under **mkfs\_hfs** for HFS behavior.

New options:

- |                              |  |
|------------------------------|--|
| <b>-F [FSType]</b>           | Specifies which file system type the command is to operate on. |
| <b>-V</b>                    | Verbose. Echoes the completed command line.                    |
| <b>-m</b>                    | Regenerates the command line used to create the file system.   |
| <b>-d</b>                    | Forces <b>mkfs</b> to process a mounted system.                |
| <b>-o [specific_options]</b> | Specifies file system specific-options, if any.                |

Changed option:

Option	Old Behavior	New Behavior	New Option To Get Old Behavior
<b>-F</b>	Check mounted file system	Specify file system type	<b>-d</b>

Changes from previous Series 700 or 800 behavior:

The following behavior is now common to both platforms:

- Default rotational delay set to zero for all disk types.

- Cylinder group summary area expanded to accommodate `fsextend` up to 4Gb.
- Prototype file supported (to specify a file to be copied to the boot block). But `mkboot(1M)` is the normal way to do this.

You will need to specify the type of the file system `mkfs` is to operate on—either on the commandline or in `/etc/fstab`. 10.x `mkfs` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

As of 10.0, this command is stored in `/usr/sbin/mkfs`.

## **mount(1M), umount(1M)**

*For 10.20:*

The `mount(1M)` and `umount(1M)` commands have been modified to:

- support Multiple File System (MFS) in a wrapped paradigm
- support large file systems (for HFS- and VxFS-specific mount commands only)
- obsolete old options for all file system types

### **Summary of Change**

Usage for `mount`:

generic usage:

```
mount [-l] [-p|-v]
mount -a [-F FStype] [-eQ]
mount [-F FStype] [-eQrV] [-o specific_options]
      {special|directory}
mount [-F FStype] [-eQrV] [-o specific_options]
      special directory
```

HFS-specific usage:

```
mount [-l] [-p|-v]
mount -a [-F hfs] [-eQ] [-f]
```

```
mount [-F hfs] [-eQrV] [-f] [-o specific_options]
      {special|directory}
mount [-F hfs] [-eQrV] [-f] [-o specific_options]
      special directory
```

CDFS-specific usage:

```
mount [-l] [-p|-v]
mount -a [-F cdfs] [-eQ]
mount [-F cdfs] [-eQrV] [-o specific_options]
      {special|directory}
mount [-F cdfs] [-eQrV] [-o specific_options]
      special directory
```

NFS-specific usage:

```
mount [-l] [-p|-v]
mount -a [-F nfs] [-eQ]
mount [-F nfs] [-eQrV] [-o specific_options]
      {host:path|directory}
mount [-F nfs] [-eQrV] [-o specific_options]
      host:path directory
```

LOFS-specific usage:

```
mount [-p|-v]
mount -a [-F lofs] [-eQ]
mount [-F lofs] [-eQrV] [-o specific_options]
      {special_directory|directory}
mount [-F lofs] [-eQrV] [-o specific_options]
      special_directory directory
```

VxFS-specific usage:

```
mount [-l] [-v|-p]
mount [-F vxfs] [-eQ] -a
mount [-F vxfs] [-eQrV]
[-o specific_options] {special | mount_point}
```

```
mount [-F vxfs] [-eQrV]
[-o specific_options] special mount_point
```

The following options are new for HFS- and VxFS-specific `mount` commands:

```
-o largefiles      Attempts to enable the creation of files
                   greater than 2 GB in size. File systems
                   have to be created or configured to enable
                   large files (see mkfs_hfs(1M) and
                   fsadm_hfs(1M)).

-o nolargefiles   Attempts to disable the creation of files
                   greater than 2 GB in size. File systems
                   have to be created or configured to disable
                   large files (see mkfs_hfs(1M) and
                   fsadm_hfs(1M)).
```

The following options are obsolete: `-u` and `-s`.

`-t type` was obsoleted at 10.0. The replacement is `-F FStype`. The pre-10.0 `interface-format` is also obsolete. Instead, use the 10.0 `interface-format`.

Usage for `umount`:

generic usage:

```
umount [-v] [-V] {special|directory}
umount -a [-F FStype] [-v]
```

HFS-specific usage:

```
umount -a [-F hfs] [-v]
umount [-v] [-V] {special|directory}
```

NFS-specific usage:

```
umount -a [-F nfs] [-h host] [-v]
umount [-v] [-V] {host:path|directory}
```

The following options are obsolete:



-u                           The default mode enforces update (or -u's) functionality.

-t *type*                    The replacement is -F *FStype*.

The location of the `mount` and `umount` commands remains in `/usr/sbin/mount` and `/usr/sbin/umount`, respectively.

### Impact

There are backward compatibility issues. The `-u` and `-s` options have been obsoleted. Also, the `-t type` option that was obsolete in 10.0 is no longer supported. It is replaced by `-F FStype`.

### Performance

There is no significant impact to the performance of the `mount` and the `umount` commands. However, the performance of these commands is dependent on the type of the file system they operate on.

### Limitations

You must have superuser access to `mount/umount` file systems using these commands.

*For 10.0:*

- Comply with SVID3
- Support multiple file systems
- New options `-F`, `-V`, `-v`, `-e`, `-o`

`mount` and `umount` now have both generic and file system-specific syntax and options. For details, see the manpages or *HP-UX Reference*, under `mount` and `umount` for generic usage, and `mount_[FStype]` and `umount_[FStype]` for file system-specific usage—for example, look under `mount_hfs(1M)` and `umount_hfs(1M)` for HFS behavior.

New options:

`-F [FStype]`                   Specifies which file system type the command is to operate on.

`-v`                            Verbose. Echoes the completed command line.

**-e** Verbose. Writes a message to the standard output indicating which file system is being mounted. (Former **-v** behavior.)

**-o [specific\_options]** Specifies file-system-specific options, if any. New mount options for HFS:

**no\_fs\_async** Use rigorous posting of file system metadata. (This is the default.)

**fs\_async** Use relaxed posting of file system metadata.

**behind** Do the writes to disk asynchronously, where possible, without waiting for completion. Still the default on Series 700 systems.

**delayed** Delay the writes to disk till the buffer needs to be reused. Still the default on Series 800 systems. **delayed** and **behind** are mutually exclusive.

Changed option (mount):

Option	Old Behavior	New Behavior	New Option To Get Old Behavior
<b>-v</b>	Indicate which file system is being mounted or unmounted	Same but also indicate file system type	<b>-e</b>

You will need to specify the type of the file system the command is to operate on—either on the command line or in `/etc/fstab`. 10.x `mount` and `umount` consist of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

## 7-84 Commands and Libraries

## **ncheck(1M)**

*For 10.20:*

For 10.20, a new option is added to the HFS version of the `ncheck` command. The `-S sector_range` option reports on files that use the specified sector(s).

With the `-S` option, `ncheck` will print a table that reports the inode and pathname associated with the specified sectors. If no inode corresponds to the given sector, `ncheck` reports on what that sector is used for.

### **Performance**

If the `-S` option is not used, the performance of `ncheck` is not impacted.

### **Compatibility**

The output of `ncheck` has not changed for the case where `-S` is not specified. If the `-S` option is specified, `ncheck` prints a new table. Because `-S` is a new option, there are no backwards compatibility issues with respect to scripts that use `ncheck`. If the `-S` option is specified, the output will be different than the output of `ncheck` in previous releases.

*For 10.0:*

- Complies with SVID3
- Supports multiple file systems

`ncheck` now has both generic and file system-specific syntax and options. For details, see the manpages or the *HP-UX Reference*, under `ncheck` for generic usage and `ncheck_[FSType]` for file system-specific usage—for example, look under `ncheck_hfs(1M)` for HFS behavior.

New options:

- `-F [FSType]` Specifies which file system type the command is to operate on.
- `-V` Verbose. Echoes the completed command line.

Changed option:

**-i [inodes]** List of inodes must be separated by commas; spaces are not allowed. This is directly contrary to 9.x and earlier usage, when the inode numbers were separated by blanks.

You will need to specify the type of the file system `ncheck` is to operate on—either on the command line or in `/etc/fstab`. 10.x `ncheck` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

As of 10.0, this command is stored in `/usr/sbin/ncheck`.

The 9.x version of the command is available on 10.x systems under `/usr/old`.

## **ncheck\_hfs(1M)**

*For 10.20:*

A new `-S sector_range` option has been added to the HFS version of the `ncheck` command. With the `-S` option, `ncheck` will print a table that reports the inode and pathname associated with the specified sectors. If no inode corresponds to the given sector, `ncheck` reports on what that sector is used for.

The output of `ncheck` has not changed if `-S` is not specified.

Because `-S` is a new option, there are no backwards compatibility issues with respect to scripts that use `ncheck`. If the `-S` option is specified, the output will be different than the output of `ncheck` in previous releases.

## **newfs(1M)**

*For 10.0:*

- Supports multiple file systems
- `/etc/disktab` is no longer default source for disk parameters

`newfs` now has both generic and file system-specific syntax and options. For details, see the manpages or the *HP-UX Reference*, under `newfs` for generic usage and `newfs_[FSType]` for file system-specific usage—for example, look under `newfs_hfs(1M)` for HFS behavior.

New options:

- F [FSType] Specifies which file system type the command is to operate on.
- V Verbose. Echoes the completed command line.
- d Tells **newfs** to disregard a mounted file system. (Former -F behavior.)
- R [swap] Reserves [swap] megabytes of disk space for boot information or swap after the end of the file system.
- B Reserves space for boot programs after the end of the file system.
- O [disk\_type] Tells **newfs** to look for the disk type in /etc/disktab. Parameters specified on the command line override corresponding values in /etc/disktab. Values not specified on the command line or in /etc/disktab are defaulted (see below).
- o [specific\_options] Specifies file system-specific options, if any.

### Changed Options and Behavior

Changed options:

Option	Old Behavior	New Behavior	New Option To Get Old Behavior
-F	Disregard mounted file system	Specify file system type	-d
disk_type	(Specified after device file for disk parameters)	(See below)	-O

In addition, the `/etc/disktab` entries supplied by HP no longer reserve swap and boot space at the end of the file system.

- Use `-0 [disk_type]` to force `newfs` to get disk geometry parameters from `/etc/disktab`.

Otherwise, `newfs` will use the defaults shown below.

- Use `-R` and `-B` to reserve swap and boot space.

As of 10.0, `/etc/disktab` is the same file on the Series 700 and S800. In the case of some disk types, you will need to choose a Series 700- or Series 800-specific entry to get the same geometry as on 9.x.

`newfs` no longer requires you to specify `disk_type` on the command line, and, by default, does not use `/etc/disktab` to get the disk geometry parameters. Instead it uses these defaults:

```
File system size:    disk size returned by DIOC_CAPACITY
                    (minus swap/boot)
Block size:         8192
Fragment size:     1024
Tracks per cylinder: calculated as shown below
Sectors per track:  calculated as shown below
RPM:                3600
```

Tracks per cylinder and sectors per track are calculated according to the size of the file system, as follows:

File System Size	Tracks/Cylinder	Sectors/Track
<= 500 MB	7	22
501 MB - 1 GB	12	28
> 1 GB	16	39

The default value for bytes per inode has increased from 2048 to 6144.

You will need to specify the type of the file system `newfs` is to operate on—either on the command line or in `/etc/fstab`. 10.x `newfs` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

As of 10.0, this command is stored in `/usr/sbin/newfs`.

## 7-88 Commands and Libraries

The 9.x version of `newfs` is available on 10.x systems as `/usr/old/etc/newfs`.

## **quot(1M)**

*For 10.0:*

- Supports multiple file systems
- New options `-V`, `-F`

`-F [FSType]` Specifies which file system type the command is to operate on.

`-V` Prints out the usage string.

You will need to specify the type of the file system `quot` is to operate on—either on the command line or in `/etc/fstab`. 10.x `quot` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

## **quotacheck(1M)**

*For 10.0:*

- Supports multiple file systems
- New options `-V`, `-F`

`-F [FSType]` Specifies which file-system type the command is to operate on.

`-V` Prints out the usage string.

## **savecore(1M)**

*For 10.10:*

`savecore` now saves system crash dumps in a new format, which has many files for each crash dump, all saved in a subdirectory. Support for subset core files containing only user or kernel page has been removed.

Additionally, the following changes have been made to `savecore(1M)`:

- Each crash dump now has its own directory, a subdirectory of `/var/adm/crash` (or another directory specified on the command line).

- The crash dump directory contains an INDEX file describing the dump, a multitude of small files (instead of one big one) containing the physical memory image, and file(s) containing the kernel. Any of these files except INDEX can be compressed.
- **savecore** no longer offers the ability to save subsets of the core dump (for example, only user and kernel pages). All parts of the dump are saved. The relevant command line options (**-i**, **-u**, **-k**, **-S**, and **-F**) is ignored.
- **savecore** always saves as much of the dump as it can given space limitations. (The previous version of **savecore** did not save anything unless the whole dump would fit.) The option **-p** is ignored.
- The optional “system” command line parameter is ignored because it is no longer needed.
- The options **-z**, **-Z**, **-s**, **-r**, and **-m** are added to control the parameters of saving a dump in the new format.
- The options **-D** and **-O** are added to allow you to dictate where **savecore** should look for the dump in special cases such as early boot dumps. The ability of **savecore** to find the dump without help in normal cases is significantly enhanced.
- Option **-l** is added to allow **savecore** to update system log files without physically saving the dump.

For more information on these changes, refer to the **savecore(1M)** and **savecore(4)** manpages.

### Impact

Any and all programs and utilities that rely on the format of a core dump will need to change to reflect the new dump format being written by **savecore**. There are no supported non-HP core dump tools, so this should not have an impact. All HP-supported tools (such as **adb**) are changed to support the new format.

The removal of support for subset core dumps may imply a slight increase in the disk space used by core dumps for those who used this functionality.

### Performance

If you have sufficient disk space that **savecore** does not perceive a need for compression, or if you explicitly disable compression, the performance of the

## 7-90 Commands and Libraries



new version of `savecore` will be unchanged from the old. If compression is enabled, performance will degrade significantly in favor of reduced disk usage.

### **Size Requirement**

A crash dump for any given system will use incrementally more disk space (for the INDEX file), although space will be more than compensated by the space savings of compression. When compression is in use, virtual memory use of `savecore` will significantly increase.

*For 10.0:*

`savecore` has one new feature affecting both the Series 700 and 800, and a group of new features on the 700 that are already supported on Series 800 9.x systems.

### **New on Both Series 700 and 800**

The corefile is no longer saved in the directory `/tmp/syscore`; the default directory as of 10.0 is `/var/adm/crash`. You can change this and other defaults by modifying variables in `/etc/rc.config.d/savecore`:

Variable Name	Default Value (Meaning)	Description	Other Valid Values
SAVECORE	1 (always save core file)	Save core or not?	0 (do not save)
SAVECORE_DIR	/var/adm/crash	Directory where core-file will be saved	(any directory; free space in file system should at least equal RAM)
SAVE_PAGES	(blank) (save complete image)	Memory pages to save	i, u, k (see "Smaller Corefile" below)

### New on Series 700

Features formerly supported only on the Series 800, but now available on the Series 700 as well, allow you to:

- Reduce system bootup time.
- Reduce the size of the corefile after a core dump.

**Faster system bootup.** 10.x `savecore` supports multiple dump devices on both the Series 800 and 700. If the primary swap device (PSD) is *not* configured as a dump device, `savecore` will start running in the background immediately; otherwise it must save the dump image in the PSD before switching to background mode.

**Smaller corefile.** You can make `savecore` retrieve pages from the dump devices selectively by using the `-u`, `-k` or `-i` options. `-u` saves user and kernel pages; `-k` saves kernel pages only; `-i` saves the entire image if there's space. See the 10.x or (9.x Series 800) manpage for details. You can change the system

defaults for these values by modifying `/etc/rc.config.d/savecore` as shown in the table above.

## **shutdown(1M)**

*For 10.10:*

- When executed on an NFSD cluster server, `shutdown(1)` can
  - shut down all the clients in the NFSD cluster
  - shut down only the cluster server and leave the clients untouched by using the `-o` option.
- When executed on the server, warning messages can be sent to users on the clients about the impending shutdown.
- Shutdown executed on clients in an NFSD cluster can be used to only reboot or halt the client. NFSD clients cannot be shut down to the single-user state.
- The option `-r` or `-h` must be entered when executing a shutdown on an NFSD client.

*For 10.0:*

- Follows new “V.4” paradigm
- `/etc/shutdown.d` directory no longer supported
- No cluster-wide shutdown: In an NFS Diskless cluster, executing `shutdown(1M)` on the cluster server *does not* shut down all cluster nodes.

The new shutdown and startup model replaces both the shutdown actions that were previously hardcoded into `shutdown(1M)` and the execution of scripts in directory `/etc/shutdown.d`.

For information about the new model, see the *HP-UX 10.0 File System Layout White Paper* and the *HP-UX System Administration Tasks* manual.

### **Changing Existing Scripts**

If you have scripts in `/etc/shutdown.d` on a 9.x system, and you want to move them to a 10.x system, you will need to migrate them to the new method. For instructions, see the section on startup and shutdown scripts in the *Upgrading from HP-UX 9.x to 10.x* manual.

## swapon(1M)

As of 10.01, **swapon** accepts arguments to the **-m**, **-r**, and **-l** flags that have units specified. The corresponding **/etc/fstab** entries are similarly changed.

Arguments to the **-r**, **-m**, and **-l** flags, if specified without units, are assumed to be in units of file system blocks. This maintains backward compatibility. However, these arguments can be suffixed with a **k** or **M**, which change the units to kilobytes or megabytes, respectively. The **res=**, **min=**, and **lim=** entries in **/etc/fstab** behave in the same manner.

---

**Note**            Although one of the command line forms for **swapon** may be marked as obsolete in the manpage, it has not been removed. It may be removed in a future release.

---

## sysdef(1M)

*For 10.0:*

- Displays ranges and units of tunable parameters
- Flags parameters which can change while system is running
- Shows boot-time and run-time values if they differ
- No longer displays swap or memory information
- Output format changed

Use **swapinfo(1M)** to get the swap information that used to be supplied by **sysdef**. Use **vmstat(1)** to get memory information.

## syslogd(1M)

*For 10.0:*

- Kernel messages logged via **/dev/klog**
- New **-D** option

New option:

**-D**            Prevent the kernel from printing messages directly onto the system console (route them through **syslogd** instead).

Kernel messages will be routed by default to **/var/adm/syslog/syslog.log** (which is the equivalent of 9.x **/usr/adm/syslog**), or to whatever other file you

have specified for system logging. If you do not want kernel messages to appear in the logfile, specify a priority of `kern.none`.

## **umount(1M)**

*For 10.20:*

See “`mount(1M)`, `umount(1M)`”.

## **volcopy(1M)**

*For 10.0:*

- Supported on the Series 700 as well as 800
- Supports multiple file systems
- New options `-F`, `-V`

New options:

- `-F [FSType]` Specifies which file-system type the command is to operate on. Required for tape.
- `-V` Verbose. Echoes the completed command line.

Except in the case of a tape, you can omit `-F` on the command line if you have defined the file-system type in `/etc/fstab`. 10.x `volcopy` consists of a “wrapper” command which runs the version of the command appropriate to the file-system type specified. See the `fs_wrapper(5)` manpage for more information.

## **whodo(1M)**

*For 10.0:*

- Complies with SVID3
- Two new options `-h` and `-l`
- Can report on a particular user

New options:

-h            Omit report header.

-l            Report in long form.

If invoked without options, `who do` produces the same result as on 9.x.

---

## Commands No Longer Supported

This section lists commands that are no longer supported as of HP-UX 10.0, and are not discussed elsewhere in this document.

Do not depend on this section in isolation; read it conjunction with the survey of changes in Chapter 4 of this document, in particular the sections on “IO Convergence” and “Kernel Convergence.” Also, refer to the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

For changes to libraries, refer to the *README for NLS in 10.01* and to “Library Routines (libc)”, and “Math Library Routines (libm, libM)” later in this chapter.

### ■ `bfs(1)`

As of 10.0, this command has been obsoleted. Use the `ed(1)` command instead. As of the 8.0 release, it can handle up to 32M characters.

### ■ `buildlang(1M)`

Use `localedef(1M)` instead; `localedef(1M)` behaves similarly, but not identically, to `buildlang`. See “`localedef(1M)`” in this document and the manpage for details. `localedef` was formerly supported on the Series 700 only; as of 10.0, it was supported on the Series 800 as well. See also the “Routines and Macros No Longer Supported” section in the *README for NLS in 10.01*.

### ■ `mkrs(1M)`

`mkrs`, formerly supported on the Series 700, is not supported as of 10.0.

There is no replacement for this command as such; to protect yourself against the contingency of a catastrophic crash (such as a failure of the root disk), you should use the “Support Media”, a tape, disk or kit which includes a utility (`copyutil`) that allows you to create a “failsafe” 9.x or 10.x disk recovery image.

- If you have an HP support contract, you should have the “Support Media”, and the manual which tells you how to use it, the *Support Media User’s Manual* (92453-90010).

Contact your Support Representative if you do not have both the “Support Media” tape or disk, and the manual.

- If you do not have a support contract, you should be aware that you have no guaranteed way of recovering your 10.x system from a catastrophic failure.

- **monitor(1M)**

`monitor`, which was an unsupported tool in pre-10.0 releases, was no longer available as of 10.0.

The HP Glance and HP Glance Plus products provide similar data, as do core HP-UX commands such as `vmstat(1)`, `iostat(1)` and `sar(1)`.

- **mkpdf(1M), pdfck(1M), pdfdiff(1M)**

Product Description Files (PDFs), and the commands that operate on them, are no longer supported.

PDFs formed part of Distributed Update and Install, which was replaced at 10.0 by a new utility, Software Distributor (SD). Though there is no exactly equivalent replacement in SD for PDFs, SD does provide similar functionality. See “Software Distributor” in Chapter 4 of this document for more information.

- **revck(1M)**

As of 10.0, this command has been obsoleted. Use the `what(1)` command instead. Hewlett-Packard has not been delivering `ref_files` since the 8.0 release.

- **rmchg(1)**

This command was never intended to be called directly, and it is not supported. You should continue to use `cdc(1)` and `rmdel(1)`.

- **uxgen(1M)**

Use `config` or `mk_kernel` instead; see the entry for “`config(1M)`” earlier in this chapter and “Kernel Convergence” in Chapter 4 for more information.

The S800 kernel configuration file is no longer supported. As of 10.0, the kernel configuration file for all systems is `/stand/system`, which is similar to the 9.x Series 700 `dfile`.



HP recommends that you use `mk_kernel(1M)`, or SAM (the menu-driven System Administration Manager) to rebuild the kernel on both Series 700 and 800 10.x systems.

---

## Library Routines (libc)

This section covers the following topics:

- For 10.20:
  - `getrlimit()`
  - `lockf()/fcntl()`
  - `ulimit()`
- For 10.10:
  - UNIX 95 Conformance
  - New Functionality
  - Incompatibilities with Previous Versions
  - Additional Changes
  - `curses(3X)`
- For 10.0:
  - SVID3 and XPG4 Conformance
  - XPG4 Base Profile Branding
  - Summary of Changes for XPG4
  - How This Affects Your Code
  - New libc Routines at 10.0
  - libc Changes to Support Threads
  - The 10.0 File System Layout and libc
  - Other Changes to libc Routines
  - libc Routines No Longer Supported

NLS (Native Language Support) changes to libc are covered in the *README for NLS in 10.01*.

### Changes for 10.20

#### **getrlimit()**

The default value that `RLIM_FSIZE` returned was `2GB - 511`. The new default is `RLIM_INFINITY`.

#### **lockf()/fcntl()**

The `lockf()/fcntl()` interfaces previously did not allow locking of the byte at offset `2GB - 1`. This byte is now lockable.

## **ulimit()**

The default `UL.FILESIZE` that a process used to get was  $(2\text{GB} - 511)/512$ . The new default is  $2\text{GB}/512$ . This value now corresponds to `INFINITY` in the kernel. Therefore, the default file size limit for a process is `RLIM_INFINITY`.

## **Changes for 10.10**

### **UNIX 95 Conformance**

As of HP-UX 10.10, HP-UX conforms to XPG4 and to the X/Open extensions to XPG4 (X/Open Portability Guide Issue 4 version 2). This means that HP-UX conforms to UNIX 95. See the individual commands for descriptions of changes.

### **New Functionality for 10.10**

- The following application programming interfaces have been added to `libc(3)`:
  - `bsd_signal`
  - `dbm_close`
  - `dbm_delete`
  - `dbm_fetch`
  - `dbm_firstkey`
  - `dbm_forder`
  - `dbm_init`
  - `dbm_nextkey`
  - `dbm_open`
  - `dbm_store`
  - `endutxent`
  - `getdtablesize`
  - `gethostid`
  - `getpagesize`
  - `getrusage`
  - `gettimeofday`
  - `getutxent`
  - `getutxid`
  - `getutxline`
  - `getwd`

- `getcontext`
- `killpg`
- `makecontext`
- `pututxline`
- `pstat_getprocvm`
- `pstat_getstable`
- `random`
- `re_comp`
- `re_exec`
- `regcmp`
- `regex`
- `setcontext`
- `setstate`
- `sighold`
- `sigignore`
- `sigrelse`
- `sigset`
- `setreuid`
- `setutxent`
- `sigaltstack`
- `siginterrupt`
- `srandom`
- `swapcontext`
- `ualarm`
- `usleep`
- `utimes`
- `waitid`
- The functions `makecontext`, `setcontext`, `swapcontext`, and `getcontext` are only available in the archived version of `libc`.
- The database management functions have been moved from `libdbm.a` and `libndbm.a` to `libc`. Both `libdbm.a`, `libndbm.a` and the corresponding shared libraries are now obsolete.
- The functions `initstate`, `setstate`, `random`, and `srandom` have been moved from the math library to `libc`.
- The SVID3 signal handling interfaces `sighold`, `sigignore`, `sigrelse`, and `sigset` have been moved from `libV3.a` to `libc`.
- If `UNIX95` is set as described in the section “UNIX95”, the fourth parameter of the function passed to the `nftw()` call is now a pass-by-reference (not

## 7-102 Commands and Libraries

a pass-by-value as in previous releases). If `UNIX95` is not set, the default behavior is the previous behavior (a pass-by-value).

- The parameter passed to `sigpause()` is now a signal number, not as mask.
- If the calling process is not already a session leader, `setpgrp()` sets the process group ID of the calling process to the process ID of the calling process. If `setpgrp()` creates a new session, the new session has no controlling terminal.
- The XPG4 behavior is activated if you set the `UNIX95` environment variable.

See the corresponding manpages for details.

### Incompatibilities with Previous Versions

To conform to `UNIX95`, the following `libc` interfaces have changed and are now incompatible with previous versions:

The default for the above interfaces is the previous HP-UX behavior, which provides compatibility for existing applications. The XPG4 behavior is an option that is activated if you set the `UNIX95` environment variable and include `/usr/bin/xpg4` in the path for executables.

- If `UNIX95` is set as described in the section “`UNIX95`”, the fourth parameter of the function passed to the `nftw()` call is now a pass-by-reference, not a pass-by-value as in previous releases. If `UNIX95` is not set, the default behavior is the previous behavior (a pass-by-value).
- The parameter passed to `sigpause()` is now a signal number, not a mask.
- The behavior for the `setpgrp()` function is as follows:
  - If the calling process is not already a session leader, `setpgrp()` sets the process group ID of the calling process to the process ID of the calling process.
  - If `setpgrp()` creates a new session, the new session has no controlling terminal.

### Additional Changes

- `getdate()` - this function now conforms to XPG4 specifications. This function now recognizes leap seconds. In addition, the `%r` and `%R` formats work with the current locale for backwards compatibility.
- `syslog()`—this function now conforms to XPG4 specifications. This function now returns `void` rather than `int`.
- `dbm_open()`—the fourth parameter, formerly an `int` is now `mode_t`.

- `getmsg()`, `putmsg()`, and `getpmsg()`—the last parameter of these functions, formerly a "long", has been changed to `int`.
- `setpriority()` and `getpriority()`—the second parameter of these functions, formerly an `int`, has been changed to `id_t`.
- `poll`—the second parameter, formerly an `int`, has been changed to `nfsd_t`.
- `tcgetsid()`—the first parameter, formerly a `pid_t`, has been changed to `int`.
- `nftwh()`—is now obsolete.

### **curses(3X)**

- Now conforms to the X/Open Curses, Issue 4 Specification, which includes support for color, new attributes, and codesets supported by `libc`.
- The library `libcurses.a` has been removed.
- `libcurses.sl` and `/usr/include/curses.h` have been replaced with a library that conforms to the X/Open Specification. However, the previous library is still available. The old header file is accessed by using the `_HP_CURSES` define on the command line with `-D_HP_CURSES`.
- Linking `.o` files that were developed for use with the old curses is no longer supported with `libcurses`. These `.o` files can be linked with `libHcurses` (use `-lHcurses`).
- You should begin movement to the new default curses library because the ability to use `libHcurses` and `-D_HP_CURSES` will eventually be removed.

## **Changes for 10.0**

### **SVID3 and XPG4 Conformance**

As of HP-UX 10.0, `libc` conforms to the X/Open Portability Guide Issue 4 (XPG4) and System V Interface Definition, third edition (SVID3).

### **XPG4 Base Profile Branding**

XPG4 is the X/Open Portability Guide, Issue 4. HP-UX 10.x conforms to the "Base Profile", which consists of:

- XPG4 Internationalized System Interfaces and Headers
- XPG4 Commands and Utilities (Historical Compatibility)
- XPG4 C Language (ANSI C)

## **7-104 Commands and Libraries**

HP-UX has passed the VSX4 test suite; HP has received certificates from X/Open certifying HP-UX's conformance to the above components and to the profile.

The next section summarizes the changes that were made at 10.0.

### Summary of Changes for XPG4

The following tables list the new features and significant bug fixes.

#### Changes to libc.

Routine	Change
=====	
catopen	New NL_CAT_LOCALE processing.
iconv*	New iconv(), iconv_open(), iconv_close() functions.
nl_langinfo	New ALT_DIGITS, ERA, ERA_D_T_FMT, ERA_T_FMT processing.
printf	New grouping specifier functionality (') to printf() family of functions.
strfmon	New strfmon() function, and new header file monetary.h.
strptime	New strptime() function.

For more information, see the "New Behavior in Replacement Routines", "iconv", and "XPG4 Conformance" sections in the *README for NLS in 10.01*.

### Changes to libM.

Routine	Change
y0/y1/yn	Bug fix: y0(0.0), y1(0.0), yn(3.0,0.0) now return HUGE_VAL.
log/log10	Bug fix: log(0.0), log10(0.0) now return ERANGE.

See “Math Library Routines (libm, libM)” later in this chapter for more information.

### Changes to Commands.

Routine	Change
tar	Bug fix: tar now handles 100 character link name properly.
type	Added exec’able version of POSIX shell built-in command.
patch	Created dummy command. Full functionality will be in next major HP-UX release (after 10.x).
talk	Created dummy command. Full functionality will be in next major HP-UX release.



## Changes to Headers.

Routine	Change
glob.h	Added new GLOB_NOSYS #define.
iconv.h	Added new function prototypes for iconv(), iconv_open() and iconv_close().
langinfo.h	Added new ALT_DIGITS, ERA, ERA_D_T_FMT, ERA_T_FMT #defines. Corrected _NL_ITEM_MAX #define.
monetary.h	New header file
nl_types.h	Added new NL_CAT_LOCALE #define. Added new strord() function prototype.
regex.h	Added new REG_ENOSYS #define.
wchar.h	Corrected wcschr(), wcsrchr() and wcwidth() prototypes.
wordexp.h	Added new WRDE_NOSYS #define.
sys/fcntl.h	Cloned definitions of S_ISUID and S_ISGID from sys/stat.h.
sys/ipc.h	Added definition of mode_t; used it for struct ipc member "mode". Hid struct ipc members "ndx" and "wait" with __ (except in _KERNEL).
sys/msg.h	Added definitions of msgqnum_t and msglen_t. Used msgqnum_t for msg_qnum, and msglen_t for msg_qbytes and msg_ts. Corrected ANSI prototypes of msgrcv() and msgsnd() to match manpage.
sys/shm.h	Added definition of shmatt_t. Used shmatt_t for struct msg members "shm_nattch" and "shm_cnattch". Corrected ANSI prototype of shmget() to match manpage.
sys/stat.h	Protected S_ISUID and S_ISGID against redefinition.
sys/time.h	Added ANSI prototype and external declaration of strptime().

Certain additional changes to HP-UX commands were made for consistency with XPG4 specifications. These changes affect all the utilities in the HP-UX CMDS-MIN fileset. They support the use of the `LC_MESSAGES` environment variable for selecting localized message catalogs. HP will make similar changes to other commands in a future release to provide a completely consistent messaging environment.

### **How This Affects Your Code**

The declarations of some functions have changed (as shown above), so programmers may see compilation warning messages in some instances. The programmer can either ignore these messages, or update the source code to conform to the new declarations. (In most cases, external function declarations in application source code can just be deleted, as these are provided in the system header files).

To get a consistent messaging environment across all HP-provided applications and utilities, set the new `LC_MESSAGES` environment variable to match the `LANG` or `LC_ALL` environment variable, since some utilities and most applications are still using `LANG` rather than `LC_MESSAGES`.

In future releases, HP will modify all HP utilities to use `LC_MESSAGES` instead of `LANG` for message catalog selection.

See the `NL_CAT_LOCALE` information in the `catopen(3C)` manpage for information on how to make this change to your own internationalized utilities and applications.

## New libc Routines at 10.0

The following routines were new at 10.0 on the Series 700, Series 800, or both.

New on Both Series 700 and S800

=====

authdes_create	key_gendes
basename	key_setsecret
authdes_getucred	lckpddf
clearenv	netname2host
dirname	netname2user
endspent	putws
getnetname	re_comp
getspent	re_exec
getspnam	remque
grantpt	setspent
host2netname	sigsend
iconv	sigsendset
iconv_close	strfmon
iconv_open	strptime
insque	tcgetsid
isastream	ulckpddf
key_decryptsession	unlockpt
key_encryptsession	user2netname
	yp_update

New on Series 700;  
already on Series 800

=====

New on Series 800;  
already on Series 700

=====

blmode	getclock
endfsgent	gettimer
getfsgdef	mktimer
getfsgent	reltimer
getfsgid	rmtimer
getfsgnam	setclock
getfsguser	
setfsgent	
setfsgfil	

For more information, see “Commercial Security” in Chapter 5 and the “New and Changed Commands and Routines” and “iconv(3C)” sections in the *README for NLS in 10.01*.

For full details of all these routines, see the corresponding manpages or section 3 of the *HP-UX Reference*.

**lckpwnf(3C), ulckpwnf(3C).** These routines have been added to conform to the SVR4 specification. They provide programmatic access to a standard locking mechanism for controlling access to the `/etc/passwd` file. Any program that modifies `/etc/passwd` should be modified to call these routines.

`lckpwnf` locks `/etc/passwd`; `ulckpwnf` unlocks it.

See also “Commercial Security” in Chapter 5.

### **libc Changes to Support Threads**

`libc` routines now support threads. Most of the routines that have been modified will continue to work in a threaded application as they did in non-threaded ones, requiring no changes on the part of a programmer porting code to a threaded environment.

But in some cases the interface had to be changed, and in these cases HP-UX 10.x provides two versions of the routine: the original version for non-threaded applications, and a new version for threaded applications. For a list of these new thread-safe routines, see “Thread-Safe Routines” later in this section.

In addition, HP has provided new routines to support locking. For a list of these, see “New libc Routines for Locking in Threaded Applications” later in this section.

A few routines behave slightly differently in a threaded as opposed to a non-threaded context; these are discussed under “libc Routines that Behave Differently in Threaded Applications” later in this section.

A small number of `libc` routines are not supported for threaded applications. For a list of these, see “libc Routines Not Supported for Threaded Applications”.

**Thread-Safe Routines.** In each case that required an interface change, HP is delivering a new, thread-safe routine alongside the existing, non-threaded routine. The name of the thread-safe version is the name of the existing routine plus the suffix `_r`; for example, the thread-safe version of `acltostr` is `acltostr_r`.

New, thread-safe routines in `libc` are:

<code>acltostr_r</code>	<code>getlogin_r</code>	<code>nrand48_r</code>
<code>asctime_r</code>	<code>getmntent_r</code>	<code>ptsname_r</code>
<code>crypt_r</code>	<code>getnetbyaddr_r</code>	<code>pututline_r</code>
<code>ctime_r</code>	<code>getnetbyname_r</code>	<code>rand_r</code>
<code>drand48_r</code>	<code>getnetent_r</code>	<code>readdir_r</code>
<code>ecvt_r</code>	<code>getprotobyname_r</code>	<code>seed48_r</code>
<code>encrypt_r</code>	<code>getprotobynumber_r</code>	<code>setgrent_r</code>
<code>endgrent_r</code>	<code>getprotoent_r</code>	<code>sethostent_r</code>
<code>endhostent_r</code>	<code>getpwent_r</code>	<code>setkey_r_r</code>
<code>endnetent_r</code>	<code>getpwnam_r</code>	<code>setlocale_r</code>
<code>endprotoent_r</code>	<code>getpwuid_r</code>	<code>setnetent_r</code>
<code>endpwent_r</code>	<code>getservbyname_r</code>	<code>setprotoent_r</code>
<code>endservent_r</code>	<code>getservbyport_r</code>	<code>setpwent_r</code>
<code>endusershell_r</code>	<code>getservent_r</code>	<code>setservent_r</code>
<code>endutent_r</code>	<code>getutid_r</code>	<code>setusershell_r</code>
<code>erand48_r</code>	<code>getutent_r</code>	<code>setutent_r</code>
<code>fcvt_r</code>	<code>getusershell_r</code>	<code>srand48_r</code>
<code>fgetgrent_r</code>	<code>getutline_r</code>	<code>strerror_r</code>
<code>fgetpwent_r</code>	<code>gmtime_r</code>	<code>strtoacl_r</code>
<code>getdate_r</code>	<code>inet_ntoa_r</code>	<code>strtoaclpatt_r</code>
<code>getdiskbyname_r</code>	<code>jrand48_r</code>	<code>strtok_r</code>
<code>getgrent_r</code>	<code>l64a_r</code>	<code>ttynam_r</code>
<code>getgrgid_r</code>	<code>lcong48_r</code>	<code>ultoa_r</code>
<code>getgrnam_r</code>	<code>localtime_r</code>	<code>ultostr_r</code>
<code>gethostbyaddr_r</code>	<code>lrand48_r</code>	<code>utmpname_r</code>
<code>gethostbyname_r</code>	<code>ltoa_r</code>	<code>wcstok_r</code>
<code>gethostent_r</code>	<code>ltostr_r</code>	
<code>getlocale_r</code>	<code>mrnd48_r</code>	

New, thread-safe routines in `libsec` are:

```
endspwent_r
getspwaid_r
getspwaid_r
getspwent_r
getspwnam_r
getspuid_r
setspwent_r
```

For more information on routines in `libsec`, refer to “Shadow Password Routines” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

**New libc Routines for Locking in Threaded Applications.** The following routines support locking in threaded applications:

```
flockfile
funlockfile
```

`flockfile` and `funlockfile` explicitly lock and unlock a *FILE* object. But a programmer may not need to incur the overhead of locking and unlocking the file on every call.

The programmer can minimize the locking overhead for a series of calls in a threaded application by using the following new routines, which ensure exclusive access to a file for a series of `_unlocked` `stdio` calls:

```
clearerr_unlocked  fseek_unlocked      putwchar_unlocked
fclose_unlocked    ftell_unlocked      putws_unlocked
feof_unlocked      fwrite_unlocked     rewind_unlocked
ferror_unlocked   getc_unlocked       setvbuf_unlocked
fflush_unlocked   getchar_unlocked    ungetc_unlocked
fgets_unlocked    getw_unlocked
fgetwc_unlocked   getwc_unlocked
fgetws_unlocked   getwchar_unlocked
fileno_unlocked   putc_unlocked
fputs_unlocked    putchar_unlocked
fputwc_unlocked   puts_unlocked
fputws_unlocked   putw_unlocked
fread_unlocked    putwc_unlocked
```

**libc Routines that Behave Differently in Threaded Applications.** The following routines have slightly different semantics when called from a multi-threaded application using a `NULL` pointer.

```
ctermid
cuserid
tmpnam
```

These routines are called with a pointer to a buffer in which the result is to be stored. For non-threaded applications, providing a `NULL` pointer indicates that the routine is to store the result in an internal static buffer and return a pointer to that internal buffer. But if a `NULL` pointer is passed to one of these routines from a multi-threaded application, the operation is not performed and a `NULL` pointer is returned.

**libc Routines Not Supported for Threaded Applications.** The following routines are not supported for use in threaded applications and have no corresponding thread-safe version.

```
authdes_getucred      getrpcbyname
authnone_create       getrpcbynumber
bindresvport          getrpccent
callrpc               gettransient
clnt_create            hcreate
clnt_pcreateerror     hdestroy
clnt_perrno           inetgr
clnt_perror           registerrpc
clnt_spcreateerror    remexportent
clnt_sperrno          setnetgrent
clnt_sperror          setrpccent
endnetgrent           svc_register
endrpcent             svc_unregister
getexportent          xprt_register
getexportopt          xprt_unregister
getnetgrent
```

## The 10.0 File System Layout and libc

Applications that were created on pre-10.0 HP-UX releases may need **transition links** to work properly on 10.x.

Transition links are a mechanism that HP is providing as part of HP-UX 10.x to help ensure binary compatibility with pre-10.0 releases; they link 9.x pathnames to their 10.x equivalents.

---

**Note** Transition links are in effect by default on all 10.x systems. (See the *Upgrading from HP-UX 9.x to 10.x* manual for more information.)

---

Applications that will make use of the transition links include those that call any of the following functions, directly or indirectly:

confstr	ftw	getusershell	sysconf
devnm	getdate	msemlock	system
dial	getfsent	nftw	tempnam
endfsent	getfsfile	popen	undial
execlp	getfsspec	setfsent	wordexp
execvp	getfstype		

## Other Changes to libc Routines

This section discusses the following routines:

- `closelog(3C)`
- `crypt(3C)`
- `fnmatch(3C)`
- `glob(3C)`, `globfree(3C)`
- `nftw(3C)`
- `openlog(3C)`
- `regcomp(3C)`, `regexec(3C)`, `regfree(3C)`, `regerr(3C)`
- `setlocale(3C)`/`iconv(3C)`
- `wordexp(3C)`, `wordfree(3C)`

See also the next section of this document, “Math Library Routines (libm, libM)”, the *README for NLS in 10.01*, and the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual for a summary of the effects of library changes on application code.

## 7-114 Commands and Libraries



**closelog(3C).** 10.x `closelog` conforms to COSE XPG4.2 (SPEC1170); it returns `void` rather than `int`.

**crypt(3C).** In the past, when the `edflag` was non-zero (requesting decryption), `encrypt()` would return an error. The decrypt capability of the `encrypt()` function is now available; you no longer need to get a special patch from HP and link in `/usr/lib/libcrypt` with your applications in order to do decryption.

**fnmatch(3C).** `fnmatch` has changed in the following ways:

- Treatment of “[.”, “[=”, and “[:” characters within bracketed expressions.

The sequences “[.”, “[=”, and “[:” are now treated as special within bracketed expressions. They must be used to represent collation characters, equivalence classes, or `ctypes`, and they must also be properly formed.

To represent the characters “[”, “[.” (period), “[=”, and “[:” (colon) within a bracketed expression, make sure you do not order the characters so as to form the above sequences.

For example, an expression for a sequence of “a” followed by “[” or “[.” (period) would be formed as follows:

```
a[. []
```

See also “`regcomp(3C)`, `regex(3C)`, `regfree(3C)`, `regerr(3C)`”

- Header `fnmatch.h`:

- The flag `FNM_QUOTE` has been added. Its meaning is the inverse of `FNM_NOESCAPE`.
- The following error messages have been added:

```
FNM_ESLASH * ? [] can't match /  
FNM_EPERIOD * ? [] can't match .
```

**glob(3C), globfree(3C).** *For 10.0:*

■ Context-dependent files (CDFs)

CDFs were no longer supported as of HP-UX 10.0. If **glob** is used against a disk containing CDF, they will look like directories.

■ Directory names in matched strings

The trailing slash (*/*) in a directory name is now included as part of the sorting key when matched strings are being sorted.

■ Header **glob.h**:

- Values for some flags and error codes have changed.
- The flag `LIB-OTHER-GLOB_QUOTE` has been added.

It creates the backslash character (`\`), which protects the next character.

Do not mix calls to 9.x **glob()** and **globfree()** with calls to their 10.x equivalents; see “Specific Guidelines for Moving Code from 9.x to 10.01” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

**nftw(3C).** 10.x **nftw** passes the constant `FTW_SLN` to the passed function *fn* whenever the target is a dangling symbolic link.

In a future release, **nftw(3C)** may be further changed for COSE XPG4.2 (SPEC1170) compliance, to pass the address of the `struct FTW` which is passed to the called function *fn*. The existing prototype,

```
extern int nftw(const char *path,
               int(*fn)(const char *, const struct stat *, int,
                       struct FTW), int depth, int flags);
```

would become:

```
extern int nftw(const char *path, int(*fn)
               (const char *, const struct stat *, int, struct FTW *),
               int depth, int flags);
```

Such a change would require source code changes to application code which calls **nftw**.

**openlog(3C).** 10.x **openlog**, like **closelog**, returns `void` rather than `int`.

**regcomp(3C), regex(3C), regfree(3C), regerr(3C).** *For 10.0:*

- Treatment of “[.”, “[=”, and “[:” characters within bracketed expressions.

See “fnmatch(3C)” earlier in this section.

- Treatment of `^` and `$` anchors in Extended Regular Expression (ERE).

The use of duplication characters (`+`, `*`) following anchors is now illegal. The error message returned by `regcomp()` is `REG_EEOL` or `REG_BOL`.

- Anchor characters incorrectly placed.

Anchor characters incorrectly placed in an expression now cause the error `REG_EEOL`.

- Flags removed:

- `_REG_ALT`
- `_REC_C_ESC`

- Header `regex.h`:

- Values for some flags and error codes have changed.

- The following error values have been removed:

`REG_ENEWSLINE` `\n` found before end of pattern and `REG_NEWLINE` flag not set

`REG_ENSUB` more than nine `\( \)` pairs or nesting level too deep

`REG_EMEM` Out of memory while matching expression

`REG_ENOSEARCH` No remembered search string

`REG_EDUOPER` Duplication operator in illegal position

`REG_ENOEXPR` No expression with `()` or on one side of an `'|'`

- The following error codes have been added:

`REG_EBOL` Anchor character `^` not at beginning of line

`REG_EEOL` Anchor characters `$` not at end of line

`REG_ECHAR` Invalid multibyte character

- The `regex_t` structure has changed. All internal data structures have been encapsulated. Only the members defined by XPG4 are visible.

### How This Affects Your Code

- Applications that have hardcoded return codes and flags may have to be modified.

But code using the standard constants defined in the `regex.h` header file should not have any problems.

- The `regex_t` data structure is larger than the 9.x version.

**setlocale(3C)/iconv(3C).** For information on `setlocale(3C)/iconv(3C)` see the `Locales` section of the *README for NLS in 10.01* (online as `/usr/lib/nls/README.nls.10.01`).

**wordexp(3C), wordfree(3C).** Change:

- The error code `WRDE_SYNTAX` in some cases may be returned in place of `WRDE_BADCHAR`.

See also “Pattern-Matching Routines Mixed in Unusual Ways” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

### libc Routines No Longer Supported

The following three categories of routines were no longer supported as of 10.0:

- HP-proprietary routines for HP-UX clusters.
- “DUX”; see “Details of Changes for NFS Diskless” in Chapter 4.
- Many HP-proprietary routines for NLS (see “Routines and Macros No Longer Supported” and “New Behavior in Replacement Routines” in the *README for NLS in 10.01*).

For more information and advice, programmers should read “Specific Guidelines for Moving Code from 9.x to 10.01” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

---

## Math Library Routines (libm, libM)

This section describes the following changes to the HP-UX math libraries for Release 10.0:

- New C Math Library Functions
- Millicode Versions of Math Library Functions
- New Locations for Math Libraries
- How This Affects Your Code

See also “Changes in Error Handling for Math Library Functions” in the “Compatibility with 9.x Releases” section of the *Upgrading from HP-UX 9.x to 10.x* manual.

### New C Math Library Functions

The C math library contains the following new functions, which are required by the COSE Common API (Spec 1170) standard:

Function	Result
=====	
<code>expm1(x)</code>	Returns <code>exp(x) - 1</code>
<code>ilogb(x)</code>	Returns the integer form of the binary exponent of the floating-point value <code>x</code>
<code>log1p(x)</code>	Returns <code>log(1 + x)</code>
<code>nextafter(x, y)</code>	Returns the next representable neighbor of <code>x</code> in the direction of <code>y</code>
<code>random()</code>	(with <code>srandom()</code> , <code>initstate()</code> , <code>setstate()</code> ) Generates a pseudo-random number
<code>remainder(x, y)</code>	Returns exact floating-point remainder as defined by IEEE standard; identical to <code>drem()</code>

The following C math library functions, also required by the COSE Common API (Spec 1170) standard, were previously provided only in the PA1.1 libraries. They are now provided in the PA1.0 libraries as well:

Function	Result
=====	
<code>acosh(x)</code>	Returns inverse hyperbolic cosine of x
<code>asinh(x)</code>	Returns inverse hyperbolic sine of x
<code>atanh(x)</code>	Returns inverse hyperbolic tangent of x
<code>cbrt(x)</code>	Returns cube root of x
<code>logb(x)</code>	Returns the exponent of x as an integer-valued double-precision number
<code>rint(x)</code>	Rounds x to integer-valued double-precision number, in the direction of the current rounding mode
<code>scalb(x, n)</code>	Returns $x \cdot (2^n)$ , computed efficiently

Most other functions previously provided only in the PA1.1 libraries are now provided in the PA1.0 libraries as well. These functions include:

- Float versions of many mathematical functions.
- Degree-valued trigonometric functions.
- A group of functions recommended by the IEEE standard, including `fpclassify`, `copysign`, and `finite`.
- The `log2` and `drem` functions (`drem` is identical to `remainder`)
- The C library versions of the `fpgetround` functions, which enable you to manipulate the floating-point control register

Manpages are provided for all these functions.

### **Millicode versions of math library functions**

Several of the most frequently used math functions are implemented in a special millicode library as well as in the standard math library. The millicode versions have a streamlined calling sequence and are usually faster than their counterparts in the standard library.

There are millicode versions for the following FORTRAN and C functions:

<code>acos</code>	<code>log</code>
<code>asin</code>	<code>log10</code>
<code>atan</code>	<code>pow</code>
<code>cos</code>	<code>sin</code>
<code>exp</code>	<code>tan</code>

There are millicode versions for the following Pascal functions:

```
arctan
cos
exp
ln
sin
```

To get the millicode versions of any of these functions, compile your program with the `+0libcalls` or the `+0aggressive` optimization option, and use the `+DA1.1` option (the default on PA1.1 systems). The `+0libcalls` option is invoked by default when you specify the optimization type `+0aggressive`; use `+0aggressive +0nolibcalls` if you want aggressive optimization without using millicode routines. The millicode versions are implemented in the library `/usr/lib/milli.a`.

The millicode versions of functions do not provide standards-compliant error-handling. This has different implications for different languages.

- C and Pascal standards specify error-handling for library functions.  
Use millicode versions in C and Pascal programs only if your program does not require standards-compliant error handling.
- FORTRAN standards specify no error-handling for math intrinsic functions.  
Using the millicode versions has no effect on standards compliance.

### **New Locations for Math Libraries**

As of Release 10.0, the standard locations for the math libraries have changed. Libraries previously located in the directory `/lib` have moved to the directory `/usr/lib`. The following libraries are affected:

Old Location	New Location
=====	
/lib/pa1.1/libm.a	/usr/lib/pa1.1/libm.a
/lib/pa1.1/libM.a	/usr/lib/pa1.1/libM.a
/lib/libm.a	/usr/lib/libm.a
/lib/libm.sl	/usr/lib/libm.sl
/lib/libM.a	/usr/lib/libM.a
/lib/libM.sl	/usr/lib/libM.sl
/lib/libp/libm.a	/usr/lib/libp/libm.a
/lib/libp/libM.a	/usr/lib/libp/libM.a

See the *HP-UX Floating-Point Guide* (B3906-90003) for details.

## How This Affects Your Code

- New functions and millicode versions:

- No effect on existing code.
- Should substantially improve the performance of applications that are recoded to make heavy use of them.

- Library pathnames:

If you explicitly specify math library pathnames in makefiles or shell scripts, you should change the pathnames from `/lib` to `/usr/lib` before running these scripts on 10.x.

If you do not have time to make the changes immediately, the scripts should still work on 10.x, because **transition links**, which link 9.x pathnames to 10.x pathnames, are in effect by default on all 10.x systems. But do not rely on these links for the long term. See the *Upgrading from HP-UX 9.x to 10.x* manual for more information, particularly “Applications” in the “Compatibility with 9.x Releases” section.



---

## Changes to Other Commands and Files

This section discusses changes to commands and files described in sections 4 and later of the *HP-UX Reference*.

### **tztab(4)**

*For 10.10:*

The European countries (Middle European (MET and MEZ), Western European (WET), and Greenwich Mean Time Zones (GMT)) have increased the length of the Daylight Savings Time period. In 1996, the change from Summertime to Wintertime will be done on the last Sunday of October instead of the last Sunday of September. Both changes to Summer and Winter times will occur at 1 AM GMT. Therefore, the `tztab` entries have to be modified.

All countries in the above time zones are affected. If your country does not follow the new European Time Change standards, manually change the time every Fall or manually change `tztab` on every Middle European, Western European, and Greenwich Mean Time Zones system to the previous entries. The previous entries for these time zones were as follows:

```
# Greenwich Mean Time, British Summer Time
GMTOST
0 3 25-31 3 1983-1984 0    BST-1
0 3 23-29 3 1985-1990 0    BST-1
0 3 25-31 3 1991-2038 0    BST-1
0 1 25-31 10 1983-1985 0    GMT0
0 1 23-29 10 1986-2038 0    GMT0

# Middle European Time, Middle European Time Daylight Savings Time
MET-1METDST
0 3 25-31 3 1983-2038 0    METDST-2
0 2 24-30 9 1983-2038 0    MET-1

# Mitteleuropaeische Zeit, Mitteleuropaeische Sommerzeit
MEZ-1MESZ
0 3 25-31 3 1983-2038 0    MESZ-2
0 2 24-30 9 1983-2038 0    MEZ-1
```

```
# Western European Time, Western European Time Daylight Savings Time
WETOWETDST
0 3 25-31 3 1983-1984 0 WETDST-1
0 3 23-29 3 1985-2038 0 WETDST-1
0 1 25-31 10 1983-1985 0 WETO
0 1 23-29 10 1986-2038 0 WETO
```

## Changes to `/etc/disktab` (`disktab(4)`)

*For 10.0:*

`/etc/disktab` has been modified to support I/O convergence. Refer to “I/O Convergence” in Chapter 4 for more information.

This file is provided in HP-UX 10.x only for backward compatibility with previous HP-UX releases. Do not use it unless you have to; use `newfs(1M)` instead. `newfs` provides defaults for `/etc/disktab` parameters.

Changes:

- A single `/etc/disktab` file is now common to the Series 700 and S800.
  - All disk types formerly defined by the Series 700- or Series 800-specific files are defined in the converged file.
- The format of the entries has been modified as follows:
  - Entries no longer contain disk sectioning information.
    - They contain information only on the “whole disk”, which is now section 0 for both Series 700 and Series 800.
  - The disk-type labels for each entry are in the format:
 

```
<vendor>_<model>[_<other_qualifiers>]
*
```

    - So, for example, the disk identified as “hp7945” in the former `disktab` file is now identified as “HP\_7945”.
  - Series 700 entries that specified boot and swap space reservations from the file system size have been removed.

Only one entry per disk type and configuration appears; the entry does not reserve swap or boot space. The 10.x `newfs` command has runstring options to reserve swap and boot space; see the entry for “`newfs(1M)`” earlier in this chapter.

Entries suffixed with `_noswap`, `_noreserve`, or a boot or swap size such as `_64MB` no longer appear.

- For disk-types whose Series 700 and Series 800 entries were not in agreement, there is now a Series 700- or Series 800-specific entry, suffixed with `_700` or `_800`.

### How This Affects Your System

- The disk-type labels you can specify are now different for most disks from what they are on 9.x systems

See the bullet on disk-label format above.

- No boot or swap space is reserved by `newfs(1M)` unless new runstring options are specified.

In 9.x, boot and swap space is reserved when certain disk-type labels are specified.

- Existing entries provided by vendors other than HP, and entries you have created yourself, may need to be modified before you can move the file from a 9.x to a 10.x system.

The syntax of the entries accepted by `newfs` and the `getdiskbyname(3C)` function has not changed, but changes to `/etc/disktab` entries may be needed because:

- `newfs` now only uses the information for section 0 (the “whole disk”); all other sections are ignored.

For the S800, the former section 2 definitions must be moved to section 0 if the entry is to continue to be used to define the “whole disk” layout.

File systems can still be created on hard partitioned layouts using `/etc/disktab` entries without sectioning information by defaulting the disk size (see below).

- Third-parties may want to follow the changes in HP's conventions for disk-type labels (see above).
- If a field is not present in `/etc/disktab`, or is less than or equal to zero, then `newfs` will provide a default value for the field.

One place you may want this is the disk size field, which, if zero or negative, will default to the actual size of the disk or disk section, as returned by a `DIOC_CAPACITY ioctl()` for the device special file.

---

**Note**            Support for `/etc/disktab` may be removed in a future release.

---

## Variables To Be Removed from `unistd(5)`

*For 10.01:*

The following `sysconf(2)` variables are still available as of 10.01, but they will be removed in the next major release (that is after the last 10.x release):

```
_SC_COLL_ELEM_MAX
_SC_PASTE_FILES_MAX
_SC_SED_PATTERN_MAX
_SC_SENDTO_MAX
_SC_SORT_LINE_MAX
```

Programs that use these variables will continue to function on 10.x systems as they did in 9.x; that is, `sysconf()` calls with these variables will return `-1` and set `errno` to `EINVAL`.