



HEWLETT
PACKARD

HP 9000
Series 800
Computers

Release Notes for HP-UX 9.0

Release Notes for HP-UX 9.0

HP 9000 Series 800



**HP Part No. B3108-90002
Printed in USA August, 1992**

E0892

Legal Notices

The information contained 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 liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

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.

Copyright Information

Copyright © Hewlett-Packard Company 1991 Copyright © AT&T, Inc. 1980, 1984 Copyright © The Regents of the University of California 1979, 1980, 1983

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

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

Printing History

New editions of this manual will incorporate all material updated since the previous edition.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.)

The manual part number changes when extensive technical changes are incorporated.

August 1992 ... Edition 1.

Contents

1. Introduction

The Release Notes and Release Notes Files	1-2
Release Notes Purpose	1-2
How to Search for Information	1-3
Description of Chapters	1-3
Chapter 2: Hardware	1-3
Chapter 3: HP-UX Topics	1-3
Chapter 4: Other Topics	1-3
Other Sources of Information	1-4
Differing Release Notes	1-5

2. Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

SPUs	2-1
New Models of Series 800	2-1
Removed Models of Series 800	2-1
Multi-Processor Support	2-1
Features	2-1
Impact	2-1
Alternatives/Compatibility	2-2
Peripherals and Devices	2-2
HPC1503A Digital Audio Tape	2-2
Functionality	2-2
Storage Capacity	2-2
Performance	2-3
CIO Restrictions	2-3
HPC1504A DDS Drive	2-4
Introduction	2-4
Functionality	2-4
Storage Capacity	2-4
Factory Switch Settings	2-4

User control of DC mode	2-4
New Console Message	2-5
Performance	2-5
Restoring Compressed Data without Decompressing	2-6
CIO Restrictions	2-6
HPC1716C Magneto-Optic Drive	2-7
Features	2-7
Functionality	2-7
Driver and File System Interactions	2-8
C1705A/M/C and Other Autochanger Products	2-8
Introduction	2-8
Autochanger Configurations Available	2-8
Summary of Change	2-9
File System Interactions	2-9
Removal of bdevsw (block) Mag Tape Support	2-9
HP-FL Disk Array (HP C2250)	2-9
Features	2-9
Impact	2-11
Performance	2-11
Alternatives and Compatibility	2-12
HP-5000 Printer	2-13
Features	2-13
Supported Functionality	2-13
Unsupported Functionality	2-14
Summary of Changes	2-14
Impact	2-14
Performance	2-15
Configuration	2-15
Multiple Dump Devices Capability	2-15
Features	2-15
Summary of Change	2-15
Impact	2-15
Performance	2-16
Alternatives/Compatibility	2-16
Configuration	2-16
Non-LVM Dump Configuration (uxgen input file)	2-16
LVM Dump Configuration	2-16
Mixing of non-LVM and LVM Dump Configuration	2-17

Size Requirement	2-17
Raw Disk Access and Immediate Reporting	2-17
Features	2-17
QIC-Format Cartridge Tape Drive	2-18
A Potential QIC Performance Problem	2-18
Cards and Interfaces	2-19
HP-PB HP-FL Device Adapter (HP28615A)	2-19
Features	2-19
Summary of Change	2-19
Impact	2-20
System Powerfail Recovery	2-20
Features	2-20
Summary of Change	2-20
Configuration	2-21

3. Topics Related to the HP-UX System

Binary Compatibility in HP-UX	3-1
Features	3-1
Summary of Change	3-1
Rules for Developers	3-2
Incompatibilities between Series 700 and Series 800	3-2
file h/buf.h	3-3
h/conf.h	3-3
h/cs80.h	3-3
h/file.h	3-3
h/framebuf.h	3-3
h/fss.h	3-4
h/hlioctl.h	3-4
h/hpibio.h	3-4
h/mman.h	3-4
h/mp.h	3-4
h/mtio.h	3-4
h/param.h	3-4
h/pregion.h	3-4
h/proc.h	3-4
h/region.h	3-5
h/scsi.h	3-5
h/spinlock.h	3-5

h/signal.h	3-5
h/sysinfo.h	3-5
h/sysmacros.h	3-5
h/user.h	3-5
h/vas.h	3-5
h/vnode.h	3-5
machine/cpu.h	3-5
machine/dmem.h	3-5
machine/hdl_region.h	3-6
machine/hdl_vas.h	3-6
machine/iodc.h	3-6
machine/param.h	3-6
machine/pcb.h	3-6
sio/iodm.h	3-6
sio/autoch.h	3-6
ufs/ino.h	3-6
ufs/inode.h	3-6
Incompatibilities in S800 between Release 9.0 and 8.06	3-7
nfs/nfs.h	3-7
dux/cct.h	3-7
h/errno.h	3-7
h/mount.h	3-7
h and mp.h	3-7
h/param.h	3-8
h/privgrp.h	3-8
dux/protocol.h	3-8
h/sem_alpha.h	3-8
h/sem_sync.h	3-8
h/shm.h	3-9
h/spinlock.h	3-9
h/vnode.h	3-9
Incompatibilities in S800 between Release 9.0 and 8.02	3-9
nfs/nfs.h	3-9
dux/cct.h	3-10
h/param.h	3-10
h/privgrp.h	3-10
dux/protocol.h	3-10
h/sem_alpha.h	3-10

h/sem_sync.h	3-11
h/spinlock.h	3-11
h/mount.h	3-11
h/vmmeter.h	3-11
h/vnode.h	3-11
Incompatibilities in S800 between Release 9.0 and 8.0	3-12
machine/cpu.h	3-12
machine/mp.h	3-12
nfs/nfs.h	3-12
dux/cct.h	3-12
h/hpibio.h	3-13
h/param.h	3-13
h/privgrp.h	3-13
dux/protocol.h	3-13
h/sem_alpha.h	3-13
h/sem_sync.h	3-14
h/shm.h	3-14
h/spinlock.h	3-14
h/vmmeter.h	3-14
h/vnode.h	3-14
Clusters, uncluster(1M)	3-15
How to Uncluster a Server	3-15
Uncluster Options	3-16
What Uncluster Does	3-16
WARNING	3-17
Optional Final Checks to Verify the Unclustering	3-17
Core Commands	3-18
Environment Variables	3-18
Localized User Interface	3-18
Summary of Change	3-18
Configuration	3-19
Size Requirement	3-19
Accounting	3-19
Features	3-19
Summary of Change	3-19
basename(1)	3-20
Features	3-20
Summary of Change	3-20

Impact	3-20
chmod(1)	3-20
Features	3-20
Summary of Changes	3-20
chown(1)	3-21
Features	3-21
cksum(1)	3-21
Summary of Change	3-21
Impact	3-22
Alternatives and Compatibility	3-22
Obsolescence	3-22
cp(1)	3-22
Features	3-22
Summary of Change	3-22
Impact	3-24
csh(1)	3-24
Features	3-24
Summary of Change	3-24
cue(1)	3-24
Support of New Terminals	3-24
cut(1)	3-25
Summary of Changes	3-25
Impacts	3-25
Alternatives	3-25
date(1)	3-25
Summary of Changes	3-25
dcopy(1)	3-26
Features	3-26
Configuration	3-26
dd(1)	3-26
Features	3-26
Summary of Change	3-27
Block	3-27
Unblock	3-27
Notrunc	3-27
Changes due to HP-FL Disk Arrays	3-27
Impact	3-27
Performance	3-27

diskinfo(1M)	3-28
dump(1M)	3-28
Features	3-28
Summary of Change	3-28
Impact	3-28
Performance	3-28
ed(1)	3-28
Summary of Change	3-28
env(1)	3-29
Summary of Change	3-29
ff(1)	3-30
Features	3-30
fold(1)	3-30
Features	3-30
Summary of Change	3-30
Impact	3-30
Performance	3-30
Alternatives and Compatibility	3-30
fsck(1M), mkfs(1M), newfs(1M)	3-31
Features	3-31
Summary of Change	3-31
Impact	3-31
getconf(1)	3-31
Features	3-31
Summary of Changes	3-32
grep(1)	3-32
Summary of Change	3-32
id(1)	3-33
ksh(1) - Korn Shell	3-33
Summary of Change	3-33
Major Defect Fixes	3-33
Alternatives and Compatibility	3-34
Size Requirement	3-34
labelit(1M)	3-34
Features	3-34
Configuration	3-34
ln(1)	3-34
Summary of Change	3-34

Impact	3-34
Alternatives and Compatibility	3-35
lpstat(1M)	3-35
Features	3-35
Summary of Change	3-35
Impact	3-35
Performance	3-35
ls(1)	3-35
Summary of Changes	3-35
Impact	3-35
mklp(1M)	3-36
Summary of Change	3-36
Alternatives	3-36
od(1)	3-36
Features	3-36
Summary of Change	3-36
pr(1)	3-37
Summary of Changes	3-37
Impacts	3-37
printf(1)	3-37
Features	3-37
Summary of Change	3-37
rm(1)	3-37
Summary of Change	3-37
sadp(1)	3-38
Features	3-38
/etc/savecore(1M)	3-38
Features	3-38
Summary of Change	3-38
Impact	3-39
Performance	3-39
Configuration	3-39
sed(1)	3-40
Summary of Change	3-40
POSIX Shell; sh-posix(1)	3-40
Features	3-40
Summary of Change	3-41
Major Defect Fixes	3-43

Impact	3-43
Alternatives and Compatibility	3-44
Size Requirement	3-44
sort(1)	3-44
Features	3-44
Summary of Changes	3-44
Performance	3-45
su(1)	3-45
swapon(1M)	3-45
Features	3-45
Summary of Change	3-45
Machines affected	3-46
Impact	3-46
swapinfo(1M)	3-47
sysdef(1M)	3-47
Features	3-47
tail(1)	3-47
Summary of Changes	3-47
Impact	3-47
touch(1)	3-47
Summary of Changes	3-47
Impact	3-48
tr(1)	3-48
Features	3-48
Summary of Change	3-48
uxgen(1M)	3-48
vi(1)	3-48
Summary of Change	3-48
volcopy(1M)	3-48
Features	3-48
Configuration	3-49
Core Library Interfaces	3-49
Interface Name	3-49
Features	3-49
Summary of Change	3-49
Impact	3-49
Performance	3-50
Alternatives and Compatibility	3-50

Libcurses	3-50
Features	3-50
Summary of Change	3-50
Impact	3-51
Performance	3-51
fnmatch(3C)	3-51
Features	3-51
Summary of Change	3-51
fprintf(3S)	3-52
Summary of Change	3-52
Impact	3-52
fscanf(3S)	3-52
Summary of Change	3-52
Impact	3-52
getgrent(3C)	3-52
Summary of Change	3-52
glob(3C)	3-52
Summary of Change	3-52
popen(3S) and pclose(3S)	3-53
Summary of Change	3-53
Impact	3-53
regerror(3C)	3-54
Summary of Change	3-54
regexec(3C)	3-54
Summary of Change	3-54
Impact	3-54
rewind(3S)	3-54
Summary of Change	3-54
Impact	3-54
sysconf(2)	3-55
Features	3-55
Summary of Changes	3-55
system(3S)	3-55
Summary of Change	3-55
Impact	3-55
telldir(3C) and seekdir(3C)	3-56
Summary of Change	3-56
Impact	3-56

wordexp(3C)	3-56
Summary of Change	3-56
wordfree(3C)	3-57
Summary of Change	3-57
limits.h and definitions of FLT_MIN, DBL_MIN	3-57
Summary of Change	3-57
Impact	3-57
Clusters	3-58
Communications	3-58
Conformance (See Standards)	3-58
Data Communications (See Communications)	3-58
DataPair	3-59
Documentation	3-59
A Document Note for Series 800 Users	3-59
Comparison of Shell Features	3-59
New Documents	3-60
DTC Device File Access Utilities (DDFA), Part Number B1014-90012	3-60
Online Help	3-60
Documentation Errata	3-61
acctcom(1M) Command	3-61
Drivers	3-62
LAN Driver	3-62
Filesets (See Partitions)	3-63
Files and File Systems	3-63
/etc/group	3-63
lseek(2)	3-63
Features	3-63
Summary of Change	3-63
Impact	3-64
Performance	3-64
Alternatives and Compatibility	3-64
Configurations Affected	3-64
Maximum File System Size	3-65
Features	3-65
Impact	3-65
Performance	3-65
Configurations Supported	3-65

Internationalization	3-65
Localized User Interface Information Series 800	3-65
Using the Bourne Shell as the Default	3-66
Kernel	3-67
HPUXBOOT	3-67
Features	3-67
Summary of Change	3-67
New Kernel Interface	3-67
Cross booting	3-67
Booting from tape	3-68
Large Capacity Disks	3-69
LIF boot	3-69
LAN boot	3-69
Logical Volume Manager	3-69
B1 File Systems	3-69
Build Procedure	3-69
Performance	3-70
Size	3-70
Obsolete Kernel Features	3-70
Features	3-70
Kernel Support for Diskless Server and Clients	3-71
Summary of Change	3-71
Impact	3-71
Alternatives	3-71
Computers Affected	3-71
Kernel Support and Series 800 Graphics	3-71
Summary of Change	3-71
Impact	3-72
Alternatives	3-72
Computers Affected	3-72
MAP Products	3-72
Summary of Change	3-72
Impact	3-72
Computers Affected	3-73
X.25	3-73
Summary of Change	3-73
Impact	3-73
Computers Affected	3-73

Logical Page Size	3-73
Summary of Change	3-73
Impact	3-74
Computers Affected	3-74
Kernel Parameters Changes	3-74
fs_async Parameter	3-74
Alternatives and Compatibility	3-75
maxdsiz, maxtsiz, and maxssiz Parameters	3-75
Summary of Change	3-75
Impact	3-75
Performance	3-75
Alternatives and Compatibility	3-75
nxtxt Parameter	3-76
bufpages Parameter	3-76
Languages (Computer)	3-77
Localization (See Internationalization)	3-77
Licenses (Product)	3-77
Linker	3-77
New Features	3-77
LP Spooler	3-79
Removed Features	3-79
HP-UX Math Libraries	3-79
Selecting Different Versions of the Math Libraries	3-80
Locations of the Math Libraries	3-82
Contents of the PA1.1 Math Libraries	3-83
Logical Volume Manager (LVM)	3-85
Features	3-86
LVM System Installation	3-87
Pre-Installed 9.0 Systems	3-87
Migrating Root Disk to LVM	3-87
Migrating Non-Root Disks to LVM	3-87
OSF Functionality Details	3-88
Logical Partitioning	3-88
Mirroring	3-88
Bad Block Relocation	3-89
HP Value Added Features	3-89
Root, Primary Swap, and Dump Logical Volumes	3-89
Booting from a Logical Volume	3-90

The Boot Disk	3-91
LIF file LABEL	3-91
Maintenance Mode	3-92
Changes to S800 File	3-92
ISL command line override to LVM volumes	3-93
Bootng from Support Tape	3-93
Disk Device Driver Changes	3-93
Offline Access to LV	3-93
Mirror Consistency Recovery Configuration	3-94
VG Import Utilities	3-94
vgcfgbackup and vgcfgrestore	3-95
Integration of LVM with <code>/usr/bin/sam</code>	3-95
Coexistence with Switchover/UX	3-95
Implementation of LVM	3-95
Terminology	3-97
LVM Command Interface	3-97
Impact	3-98
DataPair Features	3-99
Performance	3-100
Size Requirement	3-100
Memory	3-100
Kernel Support for Large Memory	3-100
Summary of Change	3-100
Impact	3-100
Computers Affected	3-100
Buffer Cache	3-101
Changes to bufpages Between 8.0 and 9.0	3-101
Larger Buffer Cache Configurations	3-101
Summary of Change	3-101
Impact	3-102
Performance	3-103
Availability of Lockable Memory on an 8-MB System	3-104
Equivalently Mapped Memory	3-104
The eqmemsize Tunable Parameter	3-104
Summary of Change	3-105
Impact	3-105
Configuration	3-106
Pseudo Swap	3-106

Features	3-106
Summary of Change	3-106
Impact	3-107
Configuration	3-107
Size Requirement	3-107
Networking	3-108
The netmemmax Tunable Parameter	3-108
Summary of Change	3-108
Impact	3-108
Configuration	3-108
Features	3-109
Summary of Change	3-109
Impact	3-109
Performance	3-109
Partitions (Includes Filesets)	3-109
New SLIP-RUN fileset	3-110
Summary of Change	3-110
Impact	3-110
XNS Fileset	3-110
NETTRACELOG Fileset	3-110
Features	3-110
Summary of Change	3-110
Impact	3-111
Performance	3-112
Alternatives and Compatibility	3-112
Obsolescence	3-112
XNS Fileset	3-112
SCSI Information	3-113
Additions to the SCSI ioctl Interface	3-113
Shared Libraries	3-113
New Features	3-113
New Routines	3-113
New Options	3-114
Additional New Routines	3-114
Additional New Options	3-114
Shared Library Global Data	3-114
Impact	3-115
Signal Change	3-115

Impact	3-115
Errno Change	3-115
Impact	3-115
Smart-Bind	3-116
Workaround	3-116
C Math Functions and Shared Libraries	3-116
Standards Information	3-117
Symbolic Debugger	3-118
System Administration Manager (SAM)	3-118
Features	3-118
Summary of Change	3-120
Impact	3-122
Performance	3-123
Alternatives and Compatibility	3-123
Some Limitations in SAM	3-123
X11 Window System (See X Graphical User Environment) . . .	3-124
X11 Windows Dev. Toolkit (See X Graphical User Environment)	3-124
X Graphical User Environment	3-124
X11 R5 Server	3-124
X11 Server	3-124
X11 Server: Compatibility with previous releases . . .	3-124
X11 Server: New graphics device support	3-125
X11 Server: Bug compatibility mode	3-125
X11 Server: Serial input device support	3-125
X11 Server: Keyboard keymaps	3-125
X11 Server: New rgb.txt color name database files . . .	3-125
X11 R5 Xlib (libX11): Device Independent Color Support .	3-126
Series 800 ONLY	3-127
X11 R5 Font Server	3-127
Features	3-127
Running the Font Server	3-128
Summary of Change	3-129
Alternatives and Compatibility	3-129
X11R5 Clients	3-129
Features	3-129
Summary of Change	3-130
Impact	3-131
Performance	3-131

Alternatives and Compatibility	3-132
X11R5 Xlib	3-132
Features	3-132
Summary of Change	3-132
Internationalization	3-132
Controlling Keyboard Input Using HP's X Window System	3-133
Mapping keyboard for both Extend-char and Meta	3-133
Dead Key Compose processing	3-135
Multi-key Compose processing	3-136
Impact	3-136
Performance	3-136
Alternatives and Compatibility	3-137
Obsolescence	3-137
X11R5 Input Extension	3-137
X11R5 Xt	3-137
Features	3-137
Summary of Change	3-138
Base Translations Resource	3-138
Resource File Search Path	3-138
Customization Resource	3-138
Per-Screen Resource Database	3-139
Impact	3-139
Performance	3-139
Alternatives and Compatibility	3-139
HP Motif 1.2	3-139
Features	3-139
Summary of Change	3-139
Drag and Drop Capability	3-140
Tear-off Menus	3-140
Performance	3-140
Insensitive Visuals	3-140
Titles for Frames	3-141
Audible Warning	3-141
Color Enhancements	3-141
Baseline Alignment	3-141
2-D Menu Traversal	3-141
Input Focus	3-142
Traversal Access Functions	3-142

Virtual Keys	3-142
Resource Management	3-142
Changes for CUA Windows Compliance	3-143
Specific Widget Changes and Enhancements	3-143
MWM Enhancements	3-143
UIL Changes	3-145
UIL Installation Note	3-145
Impact	3-146
Performance	3-146
Alternatives/compatibility	3-146
Include File Changes	3-146
Other Visual Changes	3-147
File Set Name Change	3-147
HP VUE 3.0	3-147
Installation	3-147
Dr_Vue: HP VUE Diagnostic Tool	3-147
Backward Compatibility	3-148
System Configuration Files	3-149
Mixed Networks	3-149
User Sessions	3-149
Backdrops	3-150
Resources	3-150
Front Panel	3-150
Front Panel Mail Indication	3-151
Window Management	3-151
VUE Lite	3-151
General Appearance And Behavior Changes	3-152
New VUE Components	3-152
New Courtesy Clients	3-152
VUE Library	3-153
On-Line Help	3-153
Login Manager (vuelogin)	3-153
Session Manager (vuesession)	3-154
Window/Workspace Manager (vuewm)	3-154
Style Manager	3-155
File Manager	3-155
Action and File Type Databases	3-156
Command Invoker	3-156

Toolbox	3-156
Graphical Editor	3-157
Color Icon Editor	3-157
File Annotator	3-157
4. Other Topics	
ANSI/C Compiler	4-1
C++	4-1
Features and Benefits of HP C++ 3.0	4-2
GKS	4-2
HP C/HP-UX 9.0	4-3
New HP C Compiler Features	4-3
Network File System (NFS)	4-5
Features	4-5
New Command: exportfs(1M)	4-5
New Command: automount	4-6
Enhanced Functionality: mount	4-6
Enhanced Functionality for Several Commands	4-6
Increase in NFS Limit	4-7
Impact of New Features	4-7
exportfs(1M)	4-7
mountd(1M)	4-8
HP ARPA Services: gated(1M)	4-8
Features	4-8
Impact	4-10
HP ARPA Services: DTC Utilities	4-10
Dedicated Port Configuration File	4-10
Login Access Via Telnet	4-10
DTC Device File Access (DDFA) Utilities Software Component	4-11
Dedicated Port Configuration File	4-12
Telnet Port Identification	4-13
Impact	4-13
Computer Affected	4-13
HP-UX FORTRAN/9000 9.0 Release (f77)	4-13
New Product Features	4-13
Related Documentation	4-14
Image Viewer	4-14
Wider Range of Bit-Mapped Images	4-15

Enhanced User Interface	4-16
New Sample Image Files	4-16
Starting imageview	4-16
Using imageview with HP VUE	4-17
Other Imaging Capability	4-17
Instant Ignition	4-17
The IGNITION Fileset	4-18
Boot-up Checklist	4-18
Demos	4-18
Utilities	4-18
System Information	4-18
System Administration	4-18
Printer Configuration	4-19
No-windows Login	4-19
The IGNITION-HELP fileset	4-19
Help Screens	4-19
Initial Login Welcome	4-19
Default Welcome Message	4-19
LAN/9000	4-20
LAN Driver	4-20
Support Tool Manager	4-20
Features	4-20
XSTM	4-20
MSTM	4-21
CSTM	4-21
Using the Tools	4-21
Verify operation	4-21
Diagnose operation	4-21
Exercise operation	4-22
Summary of Change	4-22
Impact	4-22
Performance	4-22
Obsolescence	4-23
Terminal I/O	4-23
New Features	4-23
Performance	4-24
LSSERVER	4-24
Enhancements to the ls_targetid utility	4-24

Corrections	4-25
NCS 1.5.1	4-26
Additions to NCS 1.5.1	4-26
NCS 1.5.1 Limitations	4-27
Using SAM to Start llbd	4-27
Enabling llbd	4-27
Disabling llbd	4-28
Restarting llbd	4-28
Kernel Regeneration Messages	4-28
NCS 1.5.1 Enhancements and Corrections	4-28
Documentation Clarifications	4-29
glbd Replication	4-29
Using rrpc_\$ Calls	4-29
NetLS	4-30
NFS/9000	4-30
Native Language Support (NLS)	4-31
Summary of New Features	4-31
WPI Routines	4-32
Character Classification	4-32
File Input and Output	4-32
Case Conversion	4-33
String Manipulation	4-33
Sorting and Collation	4-34
Impact	4-34
Alternatives	4-34
Performance	4-34
Size requirement	4-35
New Locale(1M) and Localedef(1M)	4-35
Impact	4-35
Alternatives	4-36
Performance	4-36
Size requirement	4-36
buildlang(1M)	4-36
LC_ALL Category Items	4-37
getlocale(3C)	4-37
localeconv(3C)	4-37
nlinfo(1)	4-37
nl_langinfo(3C)	4-37

setlocale(3C)	4-38
Examples:	4-38
Removal of getmsg(3C)	4-39
Background	4-39
Change	4-39
Impact	4-39
Alternatives	4-40
Size Requirement	4-40
New Locales	4-40
Impact	4-40
Alternatives	4-40
Performance data	4-40
Size requirement	4-41
Locales Modified: katakana, japanese, japanese.euc	4-41
New iconv Data Files	4-41
iso8858/1 <=> ebcdic conversion (14 west European languages, 28 files)	4-41
iso8859/2 <=> ebcdic conversion (6 east European languages, 12 files)	4-42
iso8859/5 <=> ebcdic conversion (2 Cyrillic languages, 4 files)	4-42
iso8859/6 <=> ebcdic conversion (Arabic, 2 files)	4-42
iso8859/7 <=> ebcdic conversion (Greek, 2 files)	4-42
iso8859/8 <=> ebcdic conversion (Hebrew, 2 files)	4-42
iso8859/9 <=> ebcdic conversion (Turkish, 2 files)	4-42
arabic8 <=> iso8859/6 conversion (Arabic, 2 files)	4-42
greek8 <=> iso8859/7 conversion (Greek, 2 files)	4-42
hebrew8 <=> iso8859/8 conversion (Hebrew, 2 files)	4-43
turkish8 <=> iso8859/9 conversion (Turkish, 2 files)	4-43
thai8 <=> ebcdic conversion (Thai, 2 files)	4-43
big5 <=> roc15 conversion (Chinese, 2 files)	4-43
Impact	4-43
Alternatives	4-43
Performance data	4-43
Size requirement	4-43
Iconv tables Modified (Roman8 to EBCDIC IBM3172	
Conversion	4-43
Roman8 to EBCDIC Conversion	4-44

Roman8 to EBCDIC per Language	4-44
EBCDIC to Roman8 Conversion	4-45
Alternatives	4-45
Performance data	4-45
Size requirement	4-45
EBCDIC to Roman8 code conversion for each language . .	4-45
Iconv chte tables renamed	4-46
HP-UX Proprietary Interfaces to be Obsoleted in Next Release	4-46
HP-UX Proprietary Commands and Interfaces	4-48
PortNLS	4-49
Pascal Language	4-49
Starbase (Series 800 Only)	4-49
Starbase and the 9.0 Math Libraries (Series 300/400/700/800)	4-50
Switchover/UX	4-50
Features	4-50
Summary of Change	4-51
Impact	4-51
Performance	4-51
HP-UX Symbolic Debugger 9.0 Release (xdb)	4-52
New Product Features	4-52
Related Documentation	4-52
X11 Windows Dev. Toolkit	4-53
X11 Window System	4-53

Introduction

This document (file) contains information about release 9.0 of HP-UX, which runs on Series 800 computers.

This chapter:

- Describes the release notes and the release notes files.
- Explains the purpose of the release notes.
- Explains how to search for information.
- Describes the contents of subsequent chapters.
- Points to other sources of information about HP-UX.
- Describes differences between this file and other release notes documents.

The Release Notes and Release Notes Files

The release notes are contained in a file (or files). Whether you have one release notes file or several files depends on the product or products you obtain. All release notes files exist in a directory named:

`/etc/newconfig/90RelNotes`

Under the directory named **90RelNotes**, you find a release notes file for the 9.0 HP-UX system. The name is:

`hpuxsystem`

If you add a product to your system and that product has its own release notes file, the **90RelNotes** directory will contain a release notes file for that product after you perform the update. For example, assuming you install (or update to) the 9.0 release of HP-UX and subsequently update the 9.0 version by adding C and Pascal, the `/etc/newconfig/90RelNotes` directory will contain release notes files named:

`hpuxsystem`
`C`
`Pascal`

The directory for housing release notes files is named **90RelNotes** so that update processes does not overwrite any existing release notes files.

Release Notes Purpose

The release notes give you information about new, changed, and removed functionality (features). You see the new, changed, and removed functionality in relation to moving from an 8.x release to the 9.0 release of HP-UX.

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

The release notes do not document functionality. Instead of documenting functionality, the release notes contain information about the functionality and provide pointers to documentation.

How to Search for Information

- The list of topics in the next section provides an overall picture. You see the chapters and, under them, you see strings you can use for major searches. Some strings represent large sections. You might want to scan the section and then read essential information.
- Use `vi(1)` as if to edit the release notes, and use the search capabilities of `vi`. (You can use another editor for this.)
- Use `lp(1)` to print the release notes file.
- Use `grep(1)`, and similar commands, to search on strings such as topic, product, and command names.

Description of Chapters

Chapter 2: Hardware

The chapter discusses CPUs, devices, and cards. Use the following strings for searches:

Chapter 3: HP-UX Topics

The chapter discusses major entities in the core HP-UX operating system. Use the following strings for searches:

Chapter 4: Other Topics

The chapter discusses products and entities that run on the core operating system. Depending on the bundling of products, you might need to purchase some of them.

Other Sources of Information

Besides the release notes, which focus on changes from the previous release, you have other sources of information about HP-UX and its documentation.

- **README documents**

Any product you obtain might have a README document. These documents contain information about the install/update process that might not appear in the install/update document for HP-UX. You could have several README documents. For the 9.0 release, some products that previously had README documents no longer have them. Instead, the readme information might appear in release notes files. This varies across the products you can obtain.

- **/etc/newconfig/README**

A file that lists all files in the **/etc/newconfig** directory. This file describes the files, their function, and their relationship to customizable files in the 9.0 system. The information can affect you because the files in **/etc/newconfig** are updated when you perform an update.

- **/etc/newconfig/90KNOWN_PROB**

A file that lists all known problems with the 9.0 release. The file also contains known workarounds.

- **Finding HP-UX Information**

A document that describes all the documents for HP-UX.

- **HP-UX Documentation Set**

A set of documents that describes how to use HP-UX. You get information, for example, about system administration, shells, networking, windows, programming, and so on.

Differing Release Notes

This release notes is an ASCII file in the 9.0 release of HP-UX. You might have also received, prior to performing an installation or update, a printed version of this file. Because that document was printed before this file was integrated, there might be small differences in the two versions of the release notes.

However, the printed document is sufficient for you to see the new, changed, and removed functionality before you perform an installation or update.

Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

SPUs

This section contains information about system processor units (SPUs or computers) that have been included, changed, or removed for the 9.0 release.

New Models of Series 800

- Models 887 and 897
- Model 890

Removed Models of Series 800

All post 8.0 models are supported except the 825SRX, 825 and 835CHX, 834CH, 834(5)SRX, 835(5)TurboSRX.

Multi-Processor Support

Features

As of 9.0, the same operating system supports both uni-processor and multi-processor computers.

Impact

Software developers who want to support all HP-UX platforms should ensure that their software is MP safe. This means the 9.0 software should handle a potentially higher level of concurrency than the 8.x releases.

Alternatives/Compatibility

Developers are encouraged to get the 8.06 release **Application Notes**. This document explains how to detect potential problems and how to fix them.

Peripherals and Devices

This section contains information about peripherals and devices that have been included, changed, or removed for the 9.0 release.

HPC1503A Digital Audio Tape

The HPC1503A product is a second generation Digital Audio Tape (DDS) drive without data compression capability. It is a smaller form factor replacement for the HPC1502A or HPC1512A DDS drives.

Functionality

- Synchronous SCSI Transfer for improved data transfer rates
- Fastsearch functionality (compatible with HPC1502A or HPC1512A)
- SPU and device powerfail
- HP-PB
- CIO

Functionality - NOT supported:

- Restoring compressed data written on an HPC1504A device.
- Partitions
- Synchronous CIO support using the older (SPIFI-2) SCSI card

Storage Capacity

Up to 2 GBs can be stored on a 90 meter tape (subject to streaming ability).

Performance

Up to 183 KBs sustained, although synchronous transfers can burst from 1.5 to 5 MBs.

The ability of a host system to keep the drive streaming depends on the processor speed itself, the file system, the I/O configuration, and the system load.

Here are some figures (for a 60 meter tape, 90 meter tape scales accordingly):

- 1300 MB capacity
- 123 KBs per group. This is the amount of data posted to/from the media by the device for any type of transfer
- 10266 groups per tape

When the drive writes out less than a group, data-storage capacity is lost. For example, to write 64-KB blocks and not be able to stream the tape, each group on tape contains only 64 KBs of data rather than 123 KBs. Therefore, the entire tape could hold about 650 MBs rather than the expected 1300 MBs (64-KB transfers * 10266 groups = 650-MB tape capacity). Also, if the system does not keep the drive streaming, an additional 1.3 seconds of repositioning time must be added to the time needed for each write to tape. This has a severe impact on the time it takes to write a tape.

CIO Restrictions

The use of this device (in synchronous mode) with the CIO SCSI card requires an updated card containing the SPIFI-3 chip. The older CIO SCSI card containing the SPIFI-2 chip does not work with all transfer sizes and is not supported (in synchronous mode) with an HPC1503A or HPC1504A device.

The drivers detect the presence of the SPIFI-2 chip and automatically set the device to use the lower performance asynchronous mode.

HPC1504A DDS Drive

Introduction

HPC1504A is a second generation Digital Audio Tape (DDS) drive with data compression capability. It is the first data-compressed (DC) DDS drive to be supported on HP-UX Series 800 systems.

The DC functionality of HPC1504A was incorporated into the ANSI SCSI-3 standard and therefore will be the industry standard format for DC DDS recording.

Functionality

- Synchronous SCSI Transfer for improved data transfer rates
- Write data compression
- Read and decompress DC tapes
- Fastsearch functionality (compatible with HPC1502A or HPC1512A)
- SPU and device powerfail
- HP-PB
- CIO

Storage Capacity

Without compression, up to 2 GBs can be stored on a 90 meter tape. With compression up to 8 GBs can be stored on a 90 meter tape (subject to streaming ability and data compression ratio).

Factory Switch Settings

The factory should set switch 1 and 2 at the back of the HPC1504A to select DC off at power up and DC controllable by host.

User control of DC mode

The user can put the HPC1504A into data-compression mode by selecting a device file with the `c` bit set. The `c` bit is already defined as the compression bit (see `mt(7)` in the HP-UX Reference). HP does not support mixing

2-4 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

compressed and non-compressed data on the same tape. Tapes should be written in a single mode:

1. Data-compression ON, or
2. Data-compression OFF

for the entire length of the tape.

To write tapes on a HPC1504A in non-DC mode, select a device file WITHOUT the `c` bit set.

The user (or an application) CANNOT control whether or not the HPC1504A drive decompresses data it encounters when reading from the tape. The driver always attempts to decompress the data. If the data is compressed with an algorithm that HPC1504A cannot decompress, it is not done.

The density bit is meaningless for DDS devices. Therefore, it is ignored.

The no-rewind and Berkeley-style close options can be used as usual in conjunction with the compression bit.

New Console Message

A new console message is generated for decompression exception events that occur during the use of HPC1504A. The message appears as:

```
MT lu X: decompression exception event reported.
```

A decompression event occurs when the device encounters data during a read that it cannot decompress.

Performance

Up to 732 KBs sustained, although synchronous transfers can burst from 1.5 to 5 MBs.

The compression performance is dependent on the type of data being compressed and the performance of the host computer.

Certain data types are more compressible than others with commercial data bases offering the best compression potential and source code offering the worst. A typical capacity improvement in a UNIX environment is estimated to be 2.5 times and a capacity improvement in a commercial application is estimated to be 4 times.

When compressing data, the actual writes to the tape take less time. Therefore, a faster transfer rate is required to keep the drive streaming when handling compressed data. The ability of a host system to keep the drive streaming in DC mode depends on the processor speed, the file system, the I/O configuration, and the system load.

Having the drive write out less than a group causes a loss of tape capacity. For example, while writing 64-KB blocks and not being able to stream the tape, each group on tape would only contain 64 KBs of data rather than its maximum capacity. Therefore, the entire tape could only hold a fraction of its potential capacity. If the system does not keep the drive streaming, an additional 1.3 seconds of repositioning time must be added to the time needed for each write. This has a severe impact on the time required to write a tape.

Restoring Compressed Data without Decompressing

Restoring compressed data without decompressing is not supported by the SCSI Tape DM. HP-UX does not recommend or support the use of this feature.

For the Series 700, the HPC1504A drive has a method for retrieving compressed data from a tape loaded in a drive. The method either:

1. Does not support the compression algorithm the tape was written with (i.e. HPC1504A reading a DC tape from another source), or
2. Does not support any compression algorithm but supports the reading of DC tapes.

CIO Restrictions

In synchronous mode, the CIO SCSI card requires an updated card containing the SPIFI-3 chip. The older CIO SCSI card containing the SPIFI-2 chip does not work with all transfer sizes and is not supported in synchronous mode with the HPC1504A. The drivers detect the presence of the SPIFI-2 chip and automatically set the device to use the lower-performance asynchronous mode.

HPC1716C Magneto-Optic Drive

Features

Performance, reliability and capacity features of the HPC1716C:

- 325 MBs/side (650 MBs/disk), continuous composite standard format.
- Improved seek/latency (27 ms seek, 8 ms latency).
- 1000 Kb/s read transfer rate.
- 500 Kb/s write transfer rate.
- Bursts to 5 Mb/s synchronous, 3 Mb/s asynchronous.
- 3600 RPM, meaning lower latency and faster transfer rates.
- 40,000 hrs MTBF, 400,000 MSBF (Mean Swaps Between Failure).
- Less than 2.5 seconds load and spinup time, 2 seconds spindown time.
- SCSI-2.
- Split optics (lighter optical head for faster and more accurate data access).
- DSP servo (Digital Signal Processor for faster seeks and lower error rates).
- Glitchless transceivers for improved powerfail support.
- Sector atomicity with battery backup.

Functionality

Key functionality provided with the HPC1716C:

- Multifunction capability, WORM (write once, read many) and erasable media support.
- Immediate response on writes, readahead capability for optimum performance.
- Write without erase and pre-erase support.
- Wide range of product options, superset of HP-UX 8.0 optical product offering.
- 2X sequential write performance via write without erase.

- Additional 2X write performance via immediate response.

Driver and File System Interactions

While driver support is provided in 9.0 for write without erase and WORM operations, the features are not supported in any way at the file system level. At the 9.0 release, these operations are tailored to raw mode accesses.

Release 9.0 also provides driver and file system support for immediate reporting capability in the HPC1716C. This functionality in the driver is not power-fail safe. HP recommends that devices have battery backup support if using immediate reporting. The default IR setting is OFF, and can be turned ON using the `scsictl` command.

C1705A/M/C and Other Autochanger Products

Introduction

The available mechanisms include:

- Original HP/Sony erasable MO drive - (A)
- HP/Sony multifunction (erasable/WORM) drive - (M)
- HP HPC1716C multifunction (erasable/WORM) drive - (C)

These drive mechanisms are called A, M, and C, respectively.

Autochanger Configurations Available

The following configurations of optical autochangers are available. Some products are offered with any one of the of three MO mechanisms, but some are being obsoleted.

- Stand-alone unit with one mechanism - C1701A/M/C
- Model 10 10GBs Library (1 mech/16 discs) - C1703A/M
- Model 10 10GBs Library (1 mech/16 discs) - C1708C
- Model 20 20GBs Library (2 mech/32 discs) - C1700A/M/C
- Model 60 60GBs Library (4 mech/88 discs) - C1704A/M/C
- Model 100 93GBs Library (4 mech/144 discs) - C1705A/M/C

2-8 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

Summary of Change

Prior to release 9.0, the only optical products supported were the C1701A stand-alone unit and the Model 20/A Library.

In addition to the products previously supported, the 9.0 optical product offering consists of the following:

- Stand-alone unit with one mechanism - C1701C
- Model 10 10GBs Library (1 mech/ 16 discs) - C1708C
- Model 60 60GBs Library (4 mech/ 88 discs) - C1704A/C
- Model 100 93GBs Library (4 mech/144 discs) - C1705A/C

File System Interactions

Optical library products are not supported by the Logical Volume Manager.

Removal of bdevsw (block) Mag Tape Support

The Series 800 magnetic tape driver has always been different from the Series 300/400/700 tape drivers in that it allowed limited support of block functionality via the **bdevsw** table. The **mt(7)** entry in the HP-UX Reference limited support to a few applications such as the SUPPORT Tape (and those who liked to mount a tape based read-only file system where random access performance was not important).

Since the SUPPORT Tape no longer requires this functionality, and that tape accounted for much of the complexity of the software, block support is discontinued in 9.0. This has nothing to do with **blocked** support from the devices view where all records must be the same size. That is, a blocked device can still be supported through the raw (**cdevsw**) variable-length path.

HP-FL Disk Array (HP C2250)

Features

A family of HP Fiber Optic Link (HP-FL) disk array products for HP PA-RISC multiuser computer systems features one to five removable 1.36 GB SCSI-2 disk drives per array.

Space savings are achieved because the disk arrays utilize multiple small disk drives per controller in compact cabinetry. This means up to 27.2 GBs of protected data can be stored in one 19-inch by 1.6-meter cabinet. The HP-FL interface can connect up to eight disk arrays, and the fiber-optic cable can be configured in lengths up to 500 meters from the array to the host system.

The disk array products can be connected to systems that use the HP-PB architecture by way of the HP28615A HP-PB HP-FL Device Adapter card. Through this card, the HP C2250 disk array can be configured into an independent mode or into the striped configurations. With the HP28615A HP-PB HP-FL Device Adapter card, the root, primary swap, and dump are supported on these disk array products.

The striped configurations of HP's disk array family (HP C2254HA, C2254B, C2252HA, and C2252B) are supported on the HP 9000 Series 800 computer systems.

The C2254HA and C2252HA have a dedicated data protection disk drive. If a disk in the array fails, the data protection feature allows the controller to recreate the missing data as requested, invisibly to the user, with no loss in performance. The failed disk drive can be replaced on-line while the system is running, and the array controller rebuilds the missing data on a new disk with the system still on-line and available for access. The removability feature allows for interchange of system data, backup, or special security, in addition to high availability.

Information on the disk arrays is striped (spread) across two or four SCSI-2 disk drives, depending on the capacity of the array. The striped disk drives, with cache, appear to the host as one logical device. They work in unison on every disk access, serving to increase the effective data transfer rate of the array.

The C2254B and C2252B offer the same capacities as the high availability models (C2254HA and C2252HA), but do not include the dedicated data protection disk drive. These products are offered for applications where the data protection feature is not desired. With these products, all disks are dedicated to data storage.

The above configurations are joined by the independent, multi-unit configurations of the HP disk array family (HP C2251B on HP's Series 800, Models 8x2, 8x7, and 890 HP-PB computers) when connected through

2-10 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

the HP28615A HP-PB HP-FL Device Adapter card. This configuration allows independent access to each individual disk mechanism. Independent (multi-unit) mode provides increased connectivity of logical units (lu's) for this device adapter card, thus granting increased parallelism in data access.

Impact

The use of these disk devices is the same as for any HP-FL disk device. There are considerations to be made in configurations where the disk capacity exceeds 2 GBs. This is also applies for these disks when configured in the striped modes HP C2254x and C2252x).

Section 2 on the disks cannot be used for data access when the disk's arrays are configured in the larger capacity modes because the partition is larger than the file offset addressing limitation of 2 GBs. Diagnostics and `ioctl` calls might be performed on Section 2. Sections of the media that extend beyond 2 GBs from beginning of media must not be used for root, primary swap, nor dump capabilities (Sections 2, 5, 8, 9, and 11). The sections typically used for these purposes (7, 12, and 13) have been truncated to end at 2 GBs from the beginning of the media. The `/etc/disktab` file has a diagram of the partitioning scheme.

System generation and configuration for HP-UX (when including these disk array products) is discussed in the HP-UX Installing Peripherals, Volume 1 document (B2437-90005).

Performance

The performance impact of the use of these disk array products is related to the following factors:

- physical configuration of the array (ie. striped or independent)
- block-size and alignment of data transfers
- character (raw) versus block (filesystem) access
- application characteristics and load
- distribution of data on disks

For applications that perform very large data transfers, striped configurations provide a good performance alternative. Character (raw) I/O utilizing HP

C2254x configurations that are not aligned or are not even multiples of the physical block size on the media (ie. 2048 bytes) either fails (in the CIO interconnected case) or incurs a factor of three (3) performance penalty for writes (in the HP28615A HP-PB interconnected case).

HP C2252x configurations have a physical block size of 1024 bytes.

Applications that perform random, concurrent, and small-data transfers can take advantage of the increased parallelism provided with the HP C2251B independent (multi-unit) mode of disk array products when interconnected with the HP28615A HP-PB HP-FL Device Adapter card.

Alternatives and Compatibility

Striped configurations of HP-FL disk array products can be interconnected with CIO or HP-PB HP-FL device adapter cards. Independent (multi-unit) mode is only supported on those configurations utilizing the HP28615A HP-PB HP-FL Device Adapter card. Certain characteristics are peculiar to each of these arrangements:

CIO (including HP A1749A PBA-FL) Interconnected Devices have the following features:

- Striped configurations only, not Independent.
- HP C2254x configurations fail on raw transfers that are not 2048 aligned/even multiple.
- The `mediainit` command is not supported.
- DataPair/800 is supported. The 10.0 release will be the last release to support this product.

HP28615A HP-PB HP-FL Device Adapter Interconnected Devices have the following features:

- Striped and Independent (multi-unit) modes are supported.
- Root, Primary Swap, and Dump functionalities are supported.
- HP C2254x configurations incur a performance penalty for non-2048 aligned writes.
- The `mediainit` command is supported.

2-12 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

- Marginal and Unrecoverable Block sparing is performed automatically and on-line.
- DataPair/800 is NOT supported.

HP-5000 Printer

Features

The HP-5000 Model F-100 printer is manufactured by Siemens and OEM'ed by HP. The printer is available in two electrical configurations:

- P/N C2753A: 60 HZ, 208V, 16 MB graphics memory
- P/N C2754A: 50 HZ, 230/400V, 16 MB graphics memory

Additional options available for use with the Model F-100 are:

- P/N C2763A: voltage selectable transformer kit
- P/N C2758A: 16 MB graphics memory expansion module (the printer can accommodate up to three expansion modules).

Print speed	100 ppm
Paper	fanfold
Text and graphics resolution	300 dpi
Max duty cycle	2 million linear feet per month
Print buffer	32 K
Read buffer	8 K
Data memory	4 MB
Graphics and fonts memory	16 MB (standard) 64 MB (max)
Input capacity	2500 pages
Output capacity	3500 pages
Internal fonts	20 bitmapped
Internal symbol sets	14
Downloadable fonts	128
Fonts per page	128
Print technology	light emitting diode (LED) array
Hardware interface	HP Common SCSI-2
Printer language	PCL-4

Supported Functionality

- Only Series 800 computers support the HP-5000 printer.
- The HP-5000 prints a variety of data files including ASCII text, PCL-4 escape sequences, and bit-mapped graphics.

- A user can print files using the HP-UX System V spooler and Commercial HP-UX spooler—OpenSpool.
- The printer works with both HP-PB and CIO machines.
- The host computer must dedicate a single-ended SCSI card to the printer.
- Asian language printing is supported through the NLIO module.
- The printer supports HP-UX autoconfiguration.
- Sherlock diagnostics is supported through **diag0**, the diagnostic pseudo-driver. A PC-based diagnostic is also available that uses a RS-232 interface on the printer.

Unsupported Functionality

- The HP-5000 has the capability to re-print the page at the point of power failure. This page level recovery feature is not supported.
- Asynchronous status readback refers to device status in the form of readable data. This readable status exists in addition to the SCSI hardware status. Asynchronous status readback is not supported.
- Because page level recovery and asynchronous status readback are not supported, the printer does not re-print the page at the point of power failure after a device or system power fail. The HP-5000 continues to print the document after a power fail but duplicate pages are possible.
- Asynchronous Event Notification is not supported by the 9.0 SCSI DAM.
- The 9.0 release does not support multiple devices on the same SCSI card with the HP-5000 printer.

Summary of Changes

To support the HP-5000 printer a new Device Manager was created. Some changes to the printer Logical Device Manager were also necessary.

Impact

The HP-5000 Device Manager is a self-contained module and should not impact other parts of the kernel. The DM communicates with the printer LDM and the SCSI DAM in accordance with the HP-UX I/O Services Definition.

2-14 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

The HP-5000 printer LDM changes were designed to be modular and should not impact the function or performance of the other printers that use the LDM.

Performance

The HP-5000 printer driver does not directly affect system performance. When configured, the driver does increase the size of the kernel. This might have an indirect impact on performance.

Configuration

The HP-5000 device driver must be configured into the kernel in order to use the printer. Add the statement:

```
include lpr3;
```

to the `/etc/conf/gen/S800` file and regen the kernel.

Multiple Dump Devices Capability

Features

The dump device is used to store the contents of memory in the event of a system panic. For a complete dump, the device must be at least as large as memory. As of 9.0, it is no longer necessary for all of this storage be on one contiguous disk partition. It can be spread across 32 separate devices.

Summary of Change

To accommodate a maximum memory of 2 GBs, a maximum of 32 multiple dump devices capability is supported. This new dump scheme is supported for both non-LVM and LVM configurations. Dump devices can be both non-LVM and LVM disks. For LVM, only 14 dump logical devices are supported.

The dump configuration path was redesigned to provide a more reliable dump.

Impact

To take advantage of the multiple dump devices capability, the `uxgen` input file, S800, and the LVM's Boot Data Reserve Area (BDRA) (new to 9.0) should be modified as stated below in the Configuration section.

Since this functionality requires some changes in the kernel device table's definition (in `../sio/iotree.h`), any application that references the `kern_dev_table[]` should be recompiled with the new version of the file. There should be no impact for user applications.

Performance

The performance of the single dump path was increased slightly. However, when using the multiple dump devices, the multiple invocations of IODC (I/O firmware) for each device degrades the performance of the whole dump time.

Alternatives/Compatibility

To take advantage of the multiple dump devices, new `uxgen` and `savecore` commands were created for release 9.0.

Configuration

To configure multiple dump devices, the `uxgen` input file needs to change for both LVM and non-LVM dump devices. For LVM dump devices, the BDRA (Boot Data Reserve Area) also needs to be updated using the `lvlboot(1M)` command.

Non-LVM Dump Configuration (uxgen input file)

```
console on cio_ca0.mux0 at 8.1;
root    on cio_ca0.hpib0.disc1 at 8.0.0 section 10;
swap    on cio_ca0.hpib0.disc1 at 8.0.0 section 1;
dumps   on default
         cio_ca0.hpfl0.disc2 at 8.6.3 section 7
         cio_ca0.hpfl0.disc2 at 8.6.4 section 5;
```

LVM Dump Configuration

```
console on cio_ca0.mux0 at 8.1;
root    on lvl;
swap    on lvl;
dumps   on lvl;
```

When the `lvol` keyword is specified, `uxgen` passes it along as the default. The `lvol` keyword is expanded to the appropriate logical volumes at a system boot up time from the data in the BDRA.

Mixing of non-LVM and LVM Dump Configuration

```
console on cio_ca0.mux0 at 8.1;
root    on lvol;
swap    on lvol;
dumps   on lvol
        cio_ca0.hpfl0.disc2 at 8.6.3 section 7
        cio_ca0.hpfl0.disc2 at 8.6.4 section 5;
```

When the primary swap device is shared as a dump device, it has to be the first dump device in any configuration.

Size Requirement

The cumulative size of the dump device (or devices) must be equal to or greater than the memory size to perform a successful dump.

Raw Disk Access and Immediate Reporting

Features

The `scsictl` command has an option that lets a user choose immediate reporting on SCSI disks that support this feature. If immediate reporting is on and this mode is not overridden by the operating system, then an application's `write` returns as soon as the data arrives in the drive's internal buffer without waiting for the data to get to the disk medium itself. This usually causes disk writes to go faster. However, this mode is unsuitable for applications such as databases that depend on data being on the disk at the time that a write call completes.

Applications access disks either through the file system or through the character (raw) interface. In 9.0, as well as in previous releases, the kernel observes the user's setting of immediate reporting mode for most filesystem access, although it overrides the setting for some file system structures. In previous releases, the operating system always ignored the user's setting of immediate reporting for raw access because raw access to hard disks was

normally used by database-like applications. In this release, the operating system continues to ignore immediate reporting for raw access to hard disks. However, it does observe the user's setting of this mode for raw access to Magneto-Optical disks. If these disks are used as backup devices, then immediate reporting usually makes the backup run faster.

QIC-Format Cartridge Tape Drive

The Wangtek QIC (Quarter Inch Cartridge) 5525ES is an embedded, high performance cartridge tape drive with mostly SCSI-2 features. The device supports industry standard QIC-525 format as well as QIC-320/150/120/24 formats. These features allow the drive to be used as a backup storage device and for data interchange and distribution.

A Potential QIC Performance Problem

Due to a hardware limitation with the QIC-format cartridge tape drive, which is supported on the Series 800 Model 8x7S computers, a potential performance degradation can occur if the drive must backspace more than three file marks and the file marks are spaced a large distance apart.

This problem will probably not occur for HP-supplied applications. Nor should it occur on other applications where a user employs a standard HP-UX backup utility (for example, `fbackup(1m)`, `cpio(1)`, or `tar(1)`) because these commands normally write only one file mark on the tape.

To avoid problems, an application (or recovery utility) should not move around the tape in a non-sequential manner. If this movement cannot be avoided, a data tape should contain no more than three file marks. The easiest way to create multiple file marks is to specify a software device with the `no rewind` bit set because the default behavior is to rewind the tape to BOT after the data has been written.

The `no rewind` device files leave the tape positioned at EOF instead of rewinding to BOT. The naming convention for these files is `/dev/rmt/XDn` (for example, `/dev/rmt/0mn`) where:

`X` = logical unit (lu) number of the device file
`D` = data density (L = low, M = medium, H = high)

The following example would create a backup tape with four file marks:

2-18 Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

```
find /etc -print | cpio -o > /dev/rmt/Omn  
find /usr -print | cpio -o > /dev/rmt/Omn  
find /mnt -print | cpio -o > /dev/rmt/Omn  
find /tmp -print | cpio -o > /dev/rmt/Omn
```

Cards and Interfaces

This section contains information about cards and interfaces that have been included, changed, or removed for the 9.0 release.

HP-PB HP-FL Device Adapter (HP28615A)

Features

With HP-UX 9.0, HP-FL disk drives are now supported on the HP Precision Bus (HP-PB) by the use of the HP-PB Device Adapter card. This is an alternative to the HP A1749A HP-PB PBA-FL Device Adapter. The new HP28615A HP-PB HP-FL Device Adapter is a single card that uses the HP-PB protocols for the initiation of I/O rather than the CIO protocols of the earlier device adapter.

The HP28615A HP-PB HP-FL card has less power dissipation than the HP A1749A. The new driver-set for the card makes available the capability to address individual units within multiple-unit HP-FL disk devices.

Support of boot, primary swap, and dump functionality with the HP C2254x and HP C2252x disk array products are made possible with the HP28615A HP-PB HP-FL card.

Marginal and unrecoverable blocks (sectors) on HP-FL disk media are automatically spared online by the disk driver when the disks are interconnected with the system through the HP28615A HP-PB HP-FL Device Adapter card.

Summary of Change

Processors that utilize HP-PB can connect to HP-FL disks by way of the HP28615A Device Adapter. This card is full-height, so it fills two (2) slots in

the HP-PB card cage (one above and one below). To use the device adapter card with HP-FL disks, new software modules (drivers) must be included in the HP-UX kernel. The modules (driver-set) for the use of this card consist of:

hpfl1	Manages the device adapter card.
target	Allows configuration of the disk device at its bus address.
disc4	Manages the unit (normally 0) within the disk device.

Impact

To utilize the HP28615A HP-PB HP-FL Device Adapter card, the HP-UX kernel must be generated with the appropriate modules (drivers). Use include statements for each of the modules (**hpfl1**, **target**, **disc4**) added to the **gen** file. See the HP-UX Installing Peripherals, Volume 1 (B2437-90005) document for information.

DataPair/800 and 8.0 specials are not supported by the drivers for this configuration of HP-FL disks. Similar features are provided in HP-UX 9.0 by using the Logical Volume Manager.

System Powerfail Recovery

Features

Systems with a SCSI device adapter now support system powerfail recovery.

Main memory is supported by battery backup such that, if power is lost and restored within 15 minutes, the operating system is automatically restarted and processing can resume without data loss.

Summary of Change

System powerfail recovery was disabled in release 8.0 for any system including a SCSI device adapter card. This restriction was removed in 9.0.

Configuration

The battery backup hardware option must be installed.

Topics Related to the HP-UX System

This chapter contains information about topics that relate to the HP-UX system (for example, the kernel, I/O, and standards).

Binary Compatibility in HP-UX

Features

- The user applications interoperate in customer environments of heterogeneous PA HP-UX systems (i.e S700 and S800).
- The user source code can be moved without change from one PA platform to another (for features designated as common). In general, if a feature exists on multiple platforms, it should be identical on all platforms.
- User binaries can be generated on one PA platform that runs on the other.

Summary of Change

The BE kernel header files between Series 700 and Series 800 were merged as follows:

- If the structure could be merged without large changes to .c files then they were merged. Common changes to structure were addition of fields, deletion of fields, and moving around of fields.
- If the structure could not be merged easily and if the structure was used only internally within the kernel and if the user could not get access to such structures, then the structures were enclosed within `ifdef_KERNEL`.

If the structure could not be merged and could not be enclosed within `ifdef_KERNEL`, then it was considered to be an incompatible structure.

Rules for Developers

Some developers use documented system calls and commands as listed in the HP-UX Reference. Others use unsupported calls such as `/dev/kmem`. The first type is not affected by the changes in 9.0. The second type, developers who use `/dev/kmem` to read kernel data structures, can be affected. Since some structures that were visible earlier are enclosed within `ifdef_KERNEL`, the second type of developer needs to define `_KERNEL` and `_UNSUPPORTED` to access such structures.

Incompatibilities between Series 700 and Series 800

The incompatibilities exist in different features between the Series 700 and Series 800 and in structures that are common to both but could not be merged.

The following items show features that are on the Series 700 and are not on the Series 800:

- Diskless Clusters
- Memory Mapped Files
- Graphics (bitmap displays etc)
- Fast Symlinks
- User Virtual Address Space >640MB
- IOMAP(7)
- `getpriority` and `setpriority` to support `renice(1)`
- `vfork`
- Dynamic Buffer Cache
- AES L2 features (proposed but not yet in-plan)
- SCSI Disk Array
- F/W SCSI
- SCSI 8mm 9 track

3-2 Topics Related to the HP-UX System

The following items show features that are on the Series 800 and are not on the Series 700:

- SCSI Disk Mirroring
- HP-FL Disk Mirroring
- Switchover/UX
- Powerfail
- HP-FL
- HP-FL Disk Array
- Mercedes 135ppm printer
- Fair Share Scheduler—special for AT&T.
- LVM

The following items show incompatibilities according to structures:

file h/buf.h

```
struct buf      Incompatible.
```

Some #defines are different.

h/conf.h

```
struct dumpdevt Added only for S800 to support multiple device
                dump. This feature exists only for S800.
```

h/cs80.h

Major differences due to the drivers being different.

h/file.h

```
struct file     Incompatible
```

h/framebuf.h

```
struct crt_diag_args Incompatible (S700 graphics)
```

h/fss.h

This file is related to the Fair Share Scheduler which exists only for the S800.

h/hilioctl.h

All differences relate to 8042 driver supported on S700. This driver is used for graphics applications.

h/hpibio.h

Major differences due to the drivers being different.

h/mman.h

Memory mapped changes.

h/mp.h

Has #defines to support **sys_mpctl** system call on the S800. This system call is not documented but used by AT&T. This system call does not exist for the S700.

h/mtio.h

Has #defines to translate bit definitions for the tape device minor number. This exists only for S700.

h/param.h

Has #defines for the front panel led lights on the S700.

h/preregion.h

struct preregion Incompatible.

h/proc.h

struct proc Incompatible.

Extra #defines for 9.0.

3-4 Topics Related to the HP-UX System

h/region.h

struct region Incompatible.

h/scsi.h

Major changes for S700.

h/spinlock.h

struct lock Incompatible.

h/signal.h

#define SIGDIL Incompatible. (S700 feature for graphics)

h/sysinfo.h

struct sysinfo Incompatible. (sar; not supported on S700)

h/sysmacros.h

Extra #defines for S700 drivers.

h/user.h

struct user Incompatible.

h/vas.h

struct vas Incompatible.

h/vnode.h

struct vnode Incompatible.

machine/cpu.h

Extra #defines for the Series 700 board characteristics.

machine/dmem.h

struct dmem_lpmc_type Incompatible.

machine/hdl_region.h

struct hdl_region Incompatible.

machine/hdl_vas.h

struct hdl_vas Incompatible.

machine/iodc.h

Has #define VSC_NULL_PATH Incompatible.

machine/param.h

Has #defines for memory mapped changes.

machine/pcb.h

struct pcb Incompatible.

sio/iodm.h

Has #defines for some scsi devices for S800.

sio/autoch.h

Autochanger driver major numbers are different between S700 and S800. Some #defines for minor number retrieval are also different.

ufs/ino.h

Some #defines are incompatible because of fast symlinks. This is a feature only on S700.

ufs/inode.h

struct inode Incompatible. S700 has changes for fast symlinks.

Incompatibilities in S800 between Release 9.0 and 8.06

The following items show the incompatibilities between Release 9.0 and 8.06. For most items, you see two columns far to the right. The first is for release 9.0, the second is for release 8.06. You might also see descriptive information.

nfs/nfs.h

bin	source	
comp	comp	
struct fhsize		No
struct svcfh		No
struct nfswriteargs		No
struct nfs_args		No

dux/cct.h

bin	source	
comp	comp	
struct cct		No
struct addmember		No

h/errno.h

bin	source	
comp	comp	

Has #define ENONET (deleted in 9.0) No No

h/mount.h

bin	source	
comp	comp	

Has #define MOUNT_MAXTYPE (value changed to 10) No

h and mp.h

bin	source	
comp	comp	

Some `sys_mpctl` system call options have been deleted.

h/param.h

```
bin    source
comp   comp
```

```
#define UAREA has changed.
#define USRSTACK has changed.
#define KSTACKADDR has changed.
#define USRSTACKMAX has changed.
```

h/privgrp.h

```
bin    source
comp   comp
```

```
#define privmask_word1 has changed.
#define PRIV_ALLPRIV has changed.
```

dux/protocol.h

```
bin    source
comp   comp
```

```
struct receive_request_stats    No
struct recv_ops                 No
struct xmit_ops                 No
struct protocol_stats           No
```

h/sem_alpha.h

```
bin    source
comp   comp
```

```
struct sema                     No      No
```

h/sem_sync.h

```
bin    source
comp   comp
```

3-8 Topics Related to the HP-UX System

struct sync	No	No
#define WSignal was deleted	No	No

h/shm.h

bin	source	
comp	comp	
struct shmid_ds	No	No

h/spinlock.h

bin	source	
comp	comp	
struct lock	No	

h/vnode.h

bin	source	
comp	comp	
struct vnode	No	

Incompatibilities in S800 between Release 9.0 and 8.02

The following items show the incompatibilities between Release 9.0 and 8.02. For most items, you see two columns far to the right. The first is for release 9.0, the second is for release 8.02. You might also see descriptive information.

nfs/nfs.h

bin	source	
comp	comp	
struct fhsize	No	
struct svcfh	No	No
struct nfswriteargs	No	
struct nfs_args	No	

dux/cct.h

bin	source	
comp	comp	
struct cct		No
struct addmember		No

h/param.h

bin	source	
comp	comp	
#define UAREA		No
#define KSTACKADDR		No
#define USRSTACK		No
#define USRSTACKMAX		No

h/privgrp.h

bin	source	
comp	comp	
#define PRIV_ALLPRIV	No	No

dux/protocol.h

bin	source	
comp	comp	
struct receive_request_stats		No
struct recv_ops		No
struct xmit_ops		No

h/sem_alpha.h

bin	source	
comp	comp	
struct sema		No

h/sem_sync.h

bin	source
comp	comp

struct sync	No
-------------	----

h/spinlock.h

bin	source
comp	comp

struct lock	No	No
-------------	----	----

h/mount.h

bin	source
comp	comp

struct mount_data	No
-------------------	----

h/vmmeter.h

bin	source
comp	comp

struct vmtotal	No
----------------	----

h/vnode.h

bin	source
comp	comp

struct vnode	No
--------------	----

Incompatibilities in S800 between Release 9.0 and 8.0

The following items show the incompatibilities between Release 9.0 and 8.0. For most items, you see two columns far to the right. The first is for release 9.0, the second is for release 8.0. You might also see descriptive information.

machine/cpu.h

bin	source
comp	comp

#define HASH_PDIR_RATIO (deleted)	No	No
-----------------------------------	----	----

machine/mp.h

bin	source
comp	comp

#define PGSHIFT (possible problem)		
#define HPGSHIFT (possible problem)		

nfs/nfs.h

bin	source
comp	comp

struct fhsize	No	
struct svcfh	No	No
struct nfswriteargs	No	
struct nfs_args	No	

dux/cct.h

bin	source
comp	comp

struct cct	No	
struct addmember	No	No

h/hpibio.h

This file has major changes(major portions moved within _WSIO in 9.0).

h/param.h

bin	source
comp	comp

#define UAREA	No
#define KSTACKADDR	No
#define USRSTACK	No
#define USRSTACKMAX	No

h/privgrp.h

bin	source
comp	comp

#define PRIV_ALLPRIV (deleted)	No	No
--------------------------------	----	----

dux/protocol.h

bin	source
comp	comp

struct receive_request_stats	No
struct recv_ops	No
struct xmit_ops	No
struct protocol_stats	No

h/sem_alpha.h

bin	source
comp	comp

struct sema	No	No
-------------	----	----

h/sem_sync.h

bin source
comp comp

struct sync No No

h/shm.h

bin source
comp comp

struct shmid_ds No No

h/spinlock.h

bin source
comp comp

struct lock No No

h/vmmeter.h

bin source
comp comp

struct vmtotal No

h/vnode.h

bin source
comp comp

struct vnode No

Clusters, uncluster(1M)

Since a Series 800 computer cannot be a cluster server on the 9.0 release, HP has provided an uncluster utility that provides an automated way to turn a cluster server (rootserver) back to a standalone system. The unclustering process takes 10 to 30 minutes, depending on your computer and file system.

WARNING: Be sure to backup the rootserver's files before you run the uncluster utility.

How to Uncluster a Server

Work through the following steps:

1. Log in as root.
2. Examine the **uncluster(1M)** entry in the HP-UX Reference

Become familiar with how to use the utility and how it unclusters your rootserver.

3. Single-User Mode or Multi-User Mode Unclustering:

When you are ready to uncluster a rootserver, ensure that other system users are logged off. The **wall(1)** or **cwall(1)** commands can help with this.

4. If your cluster has only one file server (the rootserver you are on), then put system in single-user mode by typing **shutdown -y 0**.

If your cluster has more than one file server, leave the affected systems in multi-user mode. However, make sure that all users of these systems are logged off.

5. To uncluster the file system(or file systems), type:

```
uncluster -i -u
```

from the root (/) directory,

The uncluster program automatically compiles **parse_cdfs.c** (if necessary) and performs other initialization and error checking. You are prompted to **Perform Task?**. Type **y Return**, the default, to continue. Typing **n Return** exits the utility. This is your only opportunity to exit.

All actions taken by **uncluster** are written to the screen and a file named `/tmp/uncluster.log`.

Uncluster Options

■ **uncluster -i -u**

Uncluster performs all initialization and error checking, but does not prompt you to **Perform Task?**. The **-i** option stands for **interactive**.

■ **uncluster -u -n** or **uncluster -i -u -n**

This is the **no reboot** flag. When the **-n** option is used, **uncluster** does not reboot the system after it has been unclustered.

This option is for debugging and should not be used unless that is your intent. Typically, you should reboot the system right away.

The **-n** flag can only be used with the **-u**, **uncluster**, option.

■ **uncluster -i -p** or **uncluster -p**

The **-p** option is for **preview mode**. No modifications of any kind take place. Uncluster shows the actions it would take in uncluster mode on the screen and writes the actions to `/tmp/uncluster.log`,

There are some slight variations that take place during unclustering that do not take place during previewing. This is because nested CDFs are restored in uncluster mode, but are not touched in preview mode.

What Uncluster Does

During operation, the uncluster program prints messages that show the names of CDFs and the steps being takes to restore the CDFs to their standalone pathnames. In addition, **uncluster** prints what it is doing to clean up after unclustering is complete. Finally, you see a reboot message.

The system is in an inconsistent state after unclustering. This is because HP-UX is still running on the cluster kernel. Therefore, you should always reboot a newly unclustered system. (Again, uncluster does this automatically unless the **-n** (no reboot) flag is specified.)

WARNING

The `uncluster` utility cannot be interrupted when it is run in uncluster mode, as opposed to preview mode. Do not override this safety mechanism by killing the uncluster process as this may leave the system in an unrecoverable state.

When your computer has rebooted, it is a standalone system.

Optional Final Checks to Verify the Unclustering

The system should successfully reboot itself. If it does not, the unclustering was not successful. In the unlikely event that the unclustering fails, take the following steps to restore your system:

1. Re-install your system using your original media and the Installing and Updating HP-UX document for your release.
2. Re-install your user files and data from the backup.

When the system reboots, work through the following steps to ensure the unclustering was successful:

1. Login and execute `getcontext` at the shell prompt. The leftmost word should read `standalone`.
2. From the root directory, type `ls -H`. The `/dev` directory, for example, should no longer be a CDF (that is, it should not list as `/dev+`).
3. Examine the `/etc/checklist` file to make sure it contains no CDF pathnames. If some are there, remove them and insert the standalone pathname(s) so the affected file system (or file systems) is mounted.
4. Review `/tmp/uncluster.log` for evidence of problems. Take any required corrective actions.

Core Commands

Except HP proprietary commands, the commands in this section were modified to adhere to the IEEE P1003.2 standard.

A command line parameter that begins with a - is treated as an option. To override this, you can specify -- to terminate the options list. For example:

```
mv -- -a b
```

moves the file -a to b.

Environment Variables

The following environment variables may affect the operation of the command or system interface. Not all environment variables affect all commands or system interfaces. HP-UX Reference entries list the environment variables that affect a specific command or system interface.

LC_ALL This variable determines the values for all locale categories. The value of the LC_ALL environment variable has precedence over any of the other environment variables starting with LC_ (LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_TIME) and the LANG environment variable.

LC_MESSAGES This variable determines the locale category for processing affirmative and negative responses and the language and cultural conventions in which messages should be written.

Localized User Interface

Summary of Change

HP-UX commands, CUE, and TSM have localized user interface.

Message catalogs of HP-UX commands are localized into 6 languages:

French, German, Japanese, Korean,

Simplified Chinese, Traditional Chinese

The CUE message catalog is localized into 3 languages:

French, German, Japanese

The TSM menus and messages are localized into 2 languages:

French, German

Selected HP-UX Reference entries in Section 1 (about 50 files) are localized into Japanese.

Configuration

The NLIO product is required to use localized user interface in Asian languages.

Size Requirement

The localized user interface requires less than 0.5 MB of disk space for each language except Japanese. The Japanese user interface requires less than 1 MB.

Accounting

Features

The limitations on the number of users and connection sessions for all system accounting commands are removed.

Summary of Change

Specifically, the following commands have their limitations removed:

- `diskusg(1M)` can now support more than 1000 users.
- `acctcon1(1M)` can now support more than 500 simultaneously connection sessions.
- `acctprc1(1M)` and `acctprc2(1M)` can now support more than 500 users and 500 sessions respectively.

basename(1)

Features

The **basename** command now conforms to POSIX.2 standard.

Summary of Change

The following changes were made to **basename** to conform to the POSIX.2 standard.

If a string passed to **basename** consists entirely of slash characters, **string** is set to a single slash character. Previously, it returned an empty string.

If **string** contains trailing slash characters, they are removed. If you pass a string that ends with a slash, it returns the string preceding the slash, after the extraction. Previously, it returned an empty string.

If the suffix operand is present and is identical to the characters after the extraction, then the suffix is returned. Previously, it returned an empty string if the suffix matched the remaining string.

Impact

The changes may impact any script that uses **basename** because **basename** returns a different string based on the cases shown above. If a string contains trailing slash characters and if the suffix matches the extraction, the result is different.

chmod(1)

Features

A new permission mode, **X** , permits conditional changes to the file's access mode.

Summary of Changes

The octal form of the mode option is now listed as obsolete. It is still supported, but symbolic modes should be used instead of octal modes.

If the who part of a symbolic mode is omitted, then the permission is modified based upon the file mode creation mask that `chmod` inherits. A permission is changed only if the corresponding bit(s) in the creation mask are clear. In past releases, when `who` was omitted, then it defaulted to User, Group, and Other.

chown(1)

Features

A file's group can be changed at the same time as its owner using the new syntax:

`group synt`

cksum(1)

The `cksum` command is a synonym for `sum -p`. The 9.0 `cksum` implements a new CRC check sum algorithm as specified in the POSIX 1003.2 Draft 11.3 standard. The new algorithm is different from the one implemented in the release 8.0 version of `cksum` (`sum -p`). Consequently, the check sum output in release 9.0 probably differs from the check sum output in release 8.0. This affects only the output of `cksum` (`sum -p`). The check sum generated by executing `sum` without options is not affected (and should be the same across the two releases).

Summary of Change

The CRC checksum generating polynomial has changed to

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$$

The previous generating polynomial was

$$x^{32} + x^7 + x^5 + x^3 + x^2 + x^1 + x^0$$

For each file processed, `cksum` writes out the checksum, number of octets, pathname into the standard output. This is a change from the previous implementation where the number of blocks used to be written out instead of the number of octets.

Impact

Due to the change in the generating polynomial, the calculated checksum differs from the one that was calculated in the older versions. Applications that use this may be impacted. Any application using the number of blocks that used to be output along with the checksum is also impacted because this was changed to the number of octets.

Alternatives and Compatibility

The number of blocks can be obtained from the number of octets by dividing the number of octets by the Block Size (512). Any application that uses this to generate the checksum and then uses it for checking purposes should have no difficulty. Those that have used the older version to generate the checksum and use this version for verifying or checking will find a mismatch. There is no workaround.

Obsolescence

HP recommends you use `cksum` command instead of `sum -p`, which has become obsolete.

cp(1)

Features

A `-R` Copy file hierarchies option was added.

Summary of Change

The `-R` and `-r` options result in the source directory tree being replicated into the destination directory. In addition to regular files and directories, FIFOs, character and block device files, and symbolic links are replicated. Only `root` is able to copy device files. All other users get an error.

Symbolic links are copied so the target points to the same location that the source did. If `filea` is linked to `fileb`, than `cp filea filex` results in `filex` being linked to `fileb`.

Hard links are copied so the hard link relationship within the source directory tree is maintained within the destination directory.

3-22 Topics Related to the HP-UX System

Given:

dir0	dira
----	----
fileZ	filex (hard linked to fileZ)
	filey (hard linked to filex)

cp -R dira dirb results in:

```
dirb/filex
dirb/filey (hard linked to dirb/filex)
```

- The **-r** option now copies special files. The **warning: skipped XXXX** messages have been removed.
- The **-p** option attempts to preserve the last modification and last access times. It preserves the UID and GID. It attempts to preserve the permission and the S_ISUID and S_ISGID bits.

The **-p** option does a **chown(2)** to set the ownership information on the destination file before it does the **chmod(2)** call. If the original file did not belong to the user doing the **cp** the **chmod(2)** fails and the permissions for the destination file are left at zero (0).

- The current **cp** command did not check for failure on various file system calls. All such calls now produce an error message if they fail.
- If there are multiple file operands and the last operand does NOT exist, it is assumed to be a directory. It is created and the other **file operands** are copied into it.

Given files A and B. If **c** does not exist **cp A B c** creates directory **c** and copy files A and B into directory **c**. If **c** exists, an error message is printed.

- If an error occurs while copying a file from the source tree, the copy operation continues with the next source tree element.
- The following error message was changed:

8.0 message: cp: %s directory

9.0 message: cp: %s directory. Need -R option.

Impact

- Recursive copies now result in special files being replicated. This change results in a copy attempt for files that previously generated warning messages. If the user is not root and an attempt is made to copy a device file, a different error message is issued than occurred at 8.0.
- Scripts checking for the `cp: %s directory` error message may need to be changed.
- Copying a file owned by another user using `cp -p` results in the file being copied, the ownership being set to the owner of the source file, and the permission bits being set to zero and an error message is issued. This results because the `chmod(2)` call fails since the user is not the owner of the file.

csh(1)

Features

The `csh` shell now supports `SIGWINCH` so that resizing a window is properly reflected. This feature is already available for Series 700 computers running 8.07.

Summary of Change

Prior to this release, when getting its input from a file, `csh` would exit immediately if it was unable to execute a command (for example, if it was unable to find the command). Now, `csh` continues and attempts to execute the remaining commands in the file. This is the behavior that seems to occur on most other systems. However, to provide backward compatibility, the user may set the environment variable `EXITONERR` to 1 to get the old behavior.

cue(1)

Support of New Terminals

CUE and TSM support of new terminals (700/60, 700/96 and 700/98) is as follows:

Since 700/96 and 700/98 are direct replacements for existing/supported terminals (700/92 and 700/94, respectively), there is no special consideration.

3-24 Topics Related to the HP-UX System

With TSM, however, users should ensure that the TERM environment variable is set to either 70092 or 70094.

The 700/60 terminals are supported only if they are in the emulation mode of and existing/supported terminal. Some examples include Wyse50 and VT100. For a complete list of TSM supported terminals please refer to the TSM User's Guide (Part Number B2351-90000). The current list of CUE supported terminals is: 2392, 2394, 700/92 and 700/94.

cut(1)

Summary of Changes

The **cut(1)** command no longer treats the backspace character in any special way. This means that the combination **char-back_space-char** is treated as 3 distinct characters. Previously, this combination was treated as a single (possibly bolded) character.

Impacts

Existing scripts relying on special treatment of back space characters imbedded in an input file to **cut(1)** do not work properly. Please see the |Alternatives section for a workaround.

Alternatives

To achieve compatible behavior for **cut(1)** when an input file contains back space characters, use other commands (for example, **fold(1)** or **col(1)**) to process back spaces, and pipe the output to **cut(1)**.

date(1)

Summary of Changes

New options are added to the **date(1)** command. The additions include option C to print Century as a decimal number, option e to print day of the month as a decimal number in a two digit field with leading space fill, and some options can be modified by the E and O modifier characters to indicate a different format. Options that can be modified with the E modifier are: c, C, x, y and

Y. Options that can be modified with the O modifier are: d, e, H, I , m, M, S, U, w, W and y.

dcopy(1)

Features

This is a new command. Copies an existing file system to a new file system and in the process compresses the file system to free the unused blocks. The user can specify the new file system size, the number of new inode blocks etc.

Configuration

This command is only for HP9000 Series 800 systems.

dd(1)

Features

The dd command handles the following new features in the HP-UX 9.0

- Addition of some options (block, noblock, notrunc):

- ☐ conv=block,conv=unblock,conv=notrunc
- ☐ HP-FL Disk Arrays disk Interface

The following shows an example of executing 'dd' on the command line with the new options:

```
dd if=INPUT-FILE of=OUTPUT-FILE cbs=NUMBER conv=block
```

```
dd if=INPUT-FILE of=OUTPUT-FILE cbs=NUMBER conv=unblock
```

```
dd if=INPUT-FILE of=OUTPUT-FILE conv=notrunc
```

- Addition of code to check for a cascade type of input or output file.

Summary of Change

The **dd** command supports a conversion option to have ASCII, EBCDIC, IBM Sync outputs. New options (block, unblock, and notrunc) have been introduced for POSIX Conformance. The functions of each are as described below

Block. Treat the input as a sequence of newline terminated or EOF terminated variable length records independent of the input block boundaries. Each record shall be converted to a record with a fixed length specified by the conversion block size (The parameter **cbs**)

Unblock. Convert fixed length records to variable length. Read a number of bytes equal to the conversion block size , delete all trailing spaces and append a newline.

Notrunc. Do not truncate the output file. Preserve blocks in the output file not explicitly written by this execution of the **dd** command.

Changes due to HP-FL Disk Arrays. HP-FL Disk Arrays (flex 4-way) disks have a partition greater than 2 GBs. If **dd** uses this section as either an input or output file, and the transfer happens to be greater than 2 GBs, **dd** fails due to reasons beyond it's scope (the kernel data structures handles file offsets in **int** type and this cannot exceed 2GBs). Since a file system cannot be created on such a disk, only a raw transfer needs to be handled. The **dd** exits if it finds that the input specifies a HP-FL Disk Arrays section (usually Section 2) and its logical block size is 2 KBs (A 4-way disk speciality)

Impact

Usage is not impacted in any way except when HP-FL Disk Arrays Flex type of disk section is specified either as the input or output file.

Performance

There is no impact on performance

diskinfo(1M)

At 9.0, this command displays size in KBs to accommodate larger disks. Previously, the command displayed size in bytes.

dump(1M)

Features

The `dump` command handles the HP-FL Disk Interface in the HP-UX 9.0.

Summary of Change

HP-FL Disk Arrays (Flex 4-way) disks have a partition greater than 2 GBs. If `dd` uses this section as either an input or output file and the transfer happens to be greater than 2 GBs, `dd` fails for reasons beyond its scope (The kernel data structures handles file offsets in type `int` and this cannot exceed 2 GBs). Since a file system cannot be created on such a disk, only a raw transfer needs to be handled. The `dd` exits if it finds that the input specifies a HP-FL Disk Arrays section (usually section 2) and its logical block size is 2KBs (A 4-way disk speciality)

Impact

Usage is impacted when an HP-FL Disk Arrays 4-way disk section is specified as an input or output file.

Performance

There is no impact on performance

ed(1)

Summary of Change

The following changes have been applied to HP-UX 9.0 ed:

- The line length support (maximum number of characters per line in a file) was increased from 512 to `LINE_MAX` (2048, defined in `/usr/include/limits.h`).

- As a result of the new line length support, the possible value of the **n** suffix in a substitute command was increased from the original value of 512 to **LINE_MAX**.
- The environmental variable **SHELL** determines the preferred command-line interpreter for use in all **!** style commands. If this variable is null or not set, **sh(1)** is used.
- The **l** (**ell**) command lists most non-printable characters as one three-digit octal number with a preceding backslash. Exceptions are backslash, alert, back-space, form-feed, carriage-return, tab and vertical tab, which are listed as ****, **\a**, **\b**, **\f**, **\r**, **\t** and **\v** respectively.
- The **l** (**ell**) command marks the end of each line with a **\$**.
- Enter a end-of-file (**Ctrl-D**) character in the command mode cannot replace the **q** or **Q** command to quit a editing session.
- It is not allowed to enter a single period in input mode as backslash, period, newline. The suggested method is to precede the period with another character and then use the substitute command to delete that character.

env(1)

Summary of Change

The syntax is:

```
env [-] [-i] [name = value] ... [command [arguments ... ]]
```

If the command is executed, the exit status of **env** is the exit status of command; otherwise, **env** exits with one of the following values:

- 0 The **env** utility completed successfully.
- 1-125 An error occurred in the **env** utility
- 126 The specified command was found, but could not be executed.
- 127 The specified command could not be found.

ff(1)

Features

This new command reads the inode list and directories specified by the device and writes the information about them to standard output assuming that the device is a file system.

fold(1)

Features

A new option: **-b** (counts bytes instead of characters)

Summary of Change

The syntax is:

```
fold [ -bsw num ] input_file
```

Specifying **-b** on the command line causes the input to be folded every **num** bytes, rather than **num** columns. The value for **num** is still specified with the **-w** option. The default value for **num** is still 80.

Impact

Because some characters do not correlate well between bytes and columns, using the **-b** option can produce unexpected results. For example, it is possible for an 80 byte string to occupy as little as 1 column (all carriage returns) or as many as 640 columns (all tabs). If the input text contains tabs or other multi-column characters (example: some NLS character sets), this option is probably not appropriate.

Performance

No performance impact.

Alternatives and Compatibility

No compatibility impact.

3-30 Topics Related to the HP-UX System

fsck(1M), mkfs(1M), newfs(1M)

Features

The default behaviors for **fsck**, **mkfs**, and **newfs** have been changed to provide higher protection for the user when the target device is in use (a mounted file system, or a swap device).

It is considered disastrous to perform **newfs**, **mkfs**, and **fsck** on a swap device. Thus, **newfs**, **mkfs**, and **fsck** stop when the target device is a swap device.

The **fsck** command now warns and prompts the user when the **-n** or **-F** option is not specified and the target device is a mounted file system. However, this only happens when the init state is not in single user mode.

The **mkfs** and **newfs** commands stop to process when the target device is in use. If option **-F** is specified, **mkfs** and **newfs** do not stop when the target device is a mounted file system.

Summary of Change

A new option was added, **-F**, to force **fsck**, **mkfs**, and **newfs** to continue to process without prompt the user when the target device is a mounted file system. In addition, the three commands do not work on a swap device.

Impact

Existing **fsck**, **mkfs**, and **newfs** scripts need to specify option **-F** to make them work on a mounted file system when they are not in init state **s**.

getconf(1)

Features

The following symbols, for which configuration values can be obtained, have been added:

BC_STRING_MAX

COLL_WEIGHTS_MAX

POSIX2_BC_BASE_MAX

POSIX2_BC_DIM_MAX

POSIX2_BC_SCALE_MAX	POSIX2_BC_STRING_MAX
POSIX2_COLL_WEIGHTS_MAX	POSIX2_C_BIND
POSIX2_C_DEV	POSIX2_EXPR_NEST_MAX
POSIX2_FORT_DEV	POSIX2_FORT_RUN
POSIX2_LINE_MAX	POSIX2_LOCALEDEF
POSIX2_RE_DUP_MAX	POSIX2_SW_DEV
POSIX2_VERSION	_POSIX_ARG_MAX
_POSIX_CHILD_MAX	_POSIX_LINK_MAX
_POSIX_MAX_CANON	_POSIX_MAX_INPUT
_POSIX_NAME_MAX	_POSIX_NGROUPS_MAX
_POSIX_OPEN_MAX	_POSIX_PIPE_BUF

Summary of Changes

The following symbols have been removed:

SENDTO_MAX	COLL_ELEM_MAX
PASTE_FILES_MAX	SED_PATTERN_MAX

These were dropped from the IEEE P1003.2 standard.

grep(1)

Summary of Change

In `-e pattern_list`, the `pattern_list` can be a newline separated list of patterns. The `-e` and `-f` options can be used simultaneously.

id(1)

The **id** command now takes a **-G** option. The output format for the **-g** option was changed. The output format now has commas between supplementary group ids. This may affect any program interpreting the output of **id**.

ksh(1) - Korn Shell

Summary of Change

The **trap** command now also works with the symbolic names of the signals. The signal names can be obtained by the **kill -l** command.

Major Defect Fixes

- ksh used to be unable to open some script files due to access failures, this problem was fixed.
- The shell builtin command **kill -CONT pid** is now fixed. It no longer causes the login shell to exit when trying to awaken a suspended process.
- The problem arising because of sending a **SIGSTOP** to a child process, that killed the non-interactive POSIX Shell was fixed.
- The scope of the typeset variables in a function is within the function call. The space used by these variables is now released once the function is finished.
- While creating symbolic links, Korn Shell no longer leaves a trailing **+** in the file name.
- Changed the exit value to be 126 for files that are found but not executable and to 127 for files not found.
- The shell used to leave the **PWD** uninitialized sometimes, especially if it was not previously set. The shell now initializes the **PWD** at every execution.
- POSIX Shell can now expand patterns like **c+*** correctly in all cases.
- A **cd** to current directory failed if **.** was a stale NFS file handle. Now, this is detected and a **cd** to **full_path_name_of .** is done.
- Jobs numbers were being truncated to 8 bits. There used to be a maximum of 2 simultaneous jobs possible. This limit was enhanced to **CHILD_MAX**,

which is a configurable parameter, and is also subject to system constraints like memory available.

Alternatives and Compatibility

The HP-UX 9.0 Korn Shell has total compatibility with HP-UX 8.0 Korn Shell. Changes have been introduced only in the areas that caused defects in the Korn Shell functionalities.

Size Requirement

With the shared libraries linkage, the Korn Shell executable requires 230 KBs of disk space.

labelit(1M)

Features

This new command provides labels for file systems. Labelit can be used to provide labels for unmounted disk file systems or file systems being copied to tape.

Configuration

This command is only for HP9000 Series 800 systems.

ln(1)

Summary of Change

The `ln(1)` command no longer overwrites an existing file by default. The `-f` option must be used to overwrite an existing file.

Impact

Trying to link to an existing file with `ln(1)` fails when the `-f` option is not given. Previously, `ln(1)` would do the link, if the destination file exists, even if the `-f` option is not specified.

Alternatives and Compatibility

Use the `-f` option, if a link to an existing file is desired.

lpstat(1M)

Features

The `lpstat` command now reports the remote machine and device for remote printers.

Summary of Change

The `lpstat -v` and `lpstat -t` commands report more information for remote printers

Impact

Users see more information on remote printers.

Performance

No impact on performance.

ls(1)

Summary of Changes

If the `-F` option is specified and the file is a FIFO file, then a bar (`|`) is put after the file name.

If both options are specified in each of the following pair: `-C` and `-l` (ell), `-m` and `-l` (ell), `-x` and `-l` (ell), `-C` and `-1` (one), `-c` and `-u`, the last option specified determines the output format.

Impact

The output format may look different, if the file list contains a FIFO file and `-F` option is specified. If multiple options are specified, the output may be different, because of the differences in precedence rules.

mklp(1M)

Summary of Change

This command is obsoleted for HP-UX 9.0

Alternatives

User should use SAM to configure printers. Using `lpadmin` commands is also available to configure printers.

od(1)

Features

A new form of the command is now being supported. The old form is now listed as obsolete. It is still supported, but the new form should be preferred for use. See the HP-UX Reference for information.

Summary of Change

The `-v` option can be used to write all data, without replacing matching output lines with an asterisk.

The `-A address_base` option is one of the characters `d`, `o`, `x`, which specify the input offset base corresponding to decimal, octal, hexadecimal respectively. The character `n` can be used to specify that the offset need not be written.

The `-j skip` option can be used to jump over `skip` bytes from the beginning of the input.

The `-N count` option can be used to format no more than `count` bytes of input. `count` is by default a decimal number unless it contains a leading `0x` or `0`, in which case it is treated as a hexadecimal number or an octal number respectively.

The `-t type_string` specifies one or more output types. The specification characters `a`, `c`, `d`, `f`, `o`, `u` and `x` specify **named character**, **character**, **signed decimal**, **floating point**, **octal**, **unsigned decimal** and **hexadecimal** respectively. The specification characters `d`, `o`, `u`, `x` can be modified by an optional `C`, `S`, `I` or `L` modifier, indicating that the conversion should be

3-36 Topics Related to the HP-UX System

applied to an item of type **character**, **short**, **integer** or **long** respectively. The specifier **f** can be similarly modified by **F**, **D** or **L**, indicating that the conversion should be applied to the type **float**, **double** or **long double** respectively. Alternatively, these specifiers can be followed by an optional unsigned integer that specifies the number of bytes to be transformed by each instance of the output type.

pr(1)

Summary of Changes

The **pr(1)** now accepts **-F** option in place of **-f**.

Impacts

The support of **-f** option is unchanged; however, for portability, the use of **-F** is recommended. The two options are entirely equivalent in functionality.

printf(1)

Features

An additional conversion specifier **b** is supported.

Summary of Change

The **b** format specifier can be used to print strings. This is like the **s** conversion specifier, except that the string may contain backslash-escape sequences, which are converted to the characters they represent.

rm(1)

Summary of Change

The syntax is:

```
rm [-f|-i] [-Rr] file ...
```

When using the command to remove a directory which the user does not have write permission on, it ask the **Remove?** question if the **-i** flag is set or if the **stdin** is a **tty**.

sadp(1)

Features

This new command is a disk access profiler and hence reports disk access location and seek distance in a tabular or histogram form.

/etc/savecore(1M)

Features

Several new features are added to **savecore** for Series 800 Release 9.0. They are added to:

- Reduce the system bootup time.
- Reduce the size of the core file size after a system core dump.

Summary of Change

In Series 800 Release 9.0, users can configure more than one dump devices on their system. This is particularly useful on a large memory configuration system which one single device cannot save the entire dump image.

To reduce the bootup time of a system with large memory configured, **savecore**, by default, runs in the background **WHENEVER POSSIBLE**. By running **savecore** in the background, the system can return to its normal execution in the shortest possible time. In Release 9.0, if the Primary Swap Device (PSD) is not used as one of the dump devices, **savecore** runs in the background immediately. If the Primary Swap Device (PSD) is configured as one of the dump device, **savecore** first saves the dump image in the PSD, followed by the rest of the dump image from all other dump devices in the background.

When **savecore** is running in the background, a file **/etc/savecore.LCK** is created to lock the devices containing the dump image. Once the device is locked, the device cannot be turned on for swapping via **swapon(1M)**. A **swapon**

3-38 Topics Related to the HP-UX System

`-a` is normally part of the bootup sequence in the `/etc/rc` file. As **savecore** finishes copying the dump image from each device, by default, it turns on the device for swapping if the dump device is also used as a swap device.

To reduce the size of a core file in Release 9.0, **savecore** can selectively retrieve pages from the dump devices by using the new `-u`, `-k` or `-i` options. See **savecore(1M)** entry in the HP-UX Reference for information.

Prior to 9.0 release, **savecore** always wrote out the core file, even when there was insufficient space on the file system. In release 9.0, **savecore** checks the space available before writing out the core. The command quits if insufficient disk space available. This is to ensure that **savecore** does not consume all the remaining disk space by saving the dump onto the file system.

Impact

Prior to Release 9.0, before **savecore** wrote out a core image, it read a number from `dirname/minfree`. The core dump was not done if the number of free 512-byte blocks on the file system that contained `dirname` was less than the number obtained from the `minfree` file. In Release 9.0, the usage of the `minfree` file is slightly different. The number obtained from `minfree` represents the minimal free 512-byte blocks on the file system after **savecore** has saved the dump. Users should specify a value in `minfree` to ensure proper execution of all processes after a system panic. The number in `minfree` should be re-evaluated if one exists prior to Release 9.0.

Performance

Users can reduce the bootup time after a system panic by configuring the dump devices carefully. If the shortest bootup time is desired, the PSD should not be used as one of the dump devices. This allows **savecore** to save the dump image in the background and let the system run in multi-user mode as soon as possible.

Configuration

These new features are currently supported on Series 800 only.

sed(1)

Summary of Change

The **sed(1)** command was changed to print three-digit octal number or the corresponding escape sequence with a preceding backslash for nonprintable characters.

The **s** command's **n** flag limit was changed to be same as **LINE_MAX** (2048).

The hold space limit was changed from 4000 to 8192 characters.

POSIX Shell; sh-posix(1)

Features

The POSIX Shell (**/bin/posix/sh**) is very similar to the Korn Shell. This shell is very close to being fully compliant with the IEEE P1003.2 recommendations. The POSIX recommendations will soon be adopted as the industry standard and HP is committed to support these standards. As an indication of HP's commitment to these evolving standards, the POSIX Shell is being made the default shell in the HP-UX 9.0 release.

The files **/etc/profile**, **/etc/csh.login** and **/etc/d.login** were changed so that the path **/bin/posix** is introduced ahead of **/bin** in the default **PATH** variable. However, the users who have set up their own **PATH** in **\$HOME** or **\$HOME/.cshrc** may not see the changes mentioned below, depending on what the **PATH** was set to.

What is different:

- The default **PATH** is set to:

```
PATH=/bin/posix:/bin:/usr/bin:/usr/contrib/bin:/usr/local/bin
```

- When **sh** is typed at a shell prompt or when a shell escape like **!** is used from some utility, the POSIX Shell may be executed instead of the Bourne Shell.
- If **sh** passes as an argument to **exec1p()** or **execvp()**, then it invokes the POSIX Shell.

3-40 Topics Related to the HP-UX System

- SAM now offers `/bin/posix/sh` as the default shell when setting up new user accounts or modifying existing ones.
- Shell scripts that do not use `#!shell_path`, run under the POSIX Shell if the POSIX Shell was used to execute the shell script.

What stays the same:

- Login shells stay the same if a user's login shell is `/bin/sh`.
- Shell scripts that use `#!shell_path` continue to run as before, invoking the shell defined in the `shell_path`.
- Bourne Shell is used when the system is in the single-user mode.

Summary of Change

The following regular builtin utilities are now added as executable versions:

<code>alias</code>	Define or display aliases
<code>bg</code>	Run jobs in the background
<code>cd</code>	Change working directory
<code>false</code>	Return false value
<code>fc</code>	Process command history list
<code>fg</code>	Run jobs in the foreground
<code>getopts</code>	Parse utility options
<code>jobs</code>	Display status of jobs in the current session
<code>kill</code>	Terminate or signal processes
<code>newgrp</code>	Change to a new group
<code>read</code>	Read a line from standard input
<code>true</code>	Return true value
<code>umask</code>	Get or set file mode creation mask
<code>unalias</code>	Remove alias definitions
<code>wait</code>	Await process completion

New features of the POSIX Shell:

Added **command** to the list of builtin utilities, and also created a separate executable version of it. The **command** utility causes the shell to treat the arguments as a simple command, suppressing the shell function lookup. The shell by default does the command lookup in the order of:

```
alias expansion::builtin_utilities::function_search::PATH search
```

A **trap** now also works with the symbolic names of the signals. The signal names can be obtained by the **kill -l** command. The output of **trap** is formatted in such a way that a shell script could use to save and then later reuse a trap if it wanted. For example:

```
save_traps=$(trap) eval $save_traps
```

Added **-a** option to **unalias** builtin utility. This option can be used to remove all the alias definitions.

Added **-p** option to **export** builtin utility. When **-p** is specified, **export** writes to the standard output the names and values of all exported variables in the following format:

```
export variable=value
```

The shell formats the output, including the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same exporting results. For example:

```
export -p >temp_file
temp_file      # restores variables.
```

Added a **-p** option to **readonly** builtin utility. When **-p** is specified, **readonly** writes to the standard output, the names and values of all read-only variables, in the following format:

```
readonly constant=value
```

The shell formats the output, including the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same attribute setting results.

Added the **-f** and **-v** options to the **unset** builtin utility. If **-v** followed by a variable name is specified, the shell unsets the variable and removes it from the environment. If **-f** followed by a function name is specified, the shell unsets the function definition. If neither **-f** nor **-v** is specified, the shell checks first for a

variable name, and if the variable is not found, the shell checks for the function definition.

Added the **-S** option to the **umask** builtin utility to print the mask in symbolic mode. The default octal representation of the umask value is obsolete. For example:

```
$ umask -S
u=rwx,g=rwx,o=rx
```

Added the **-e** option to the **test** builtin command. This option returns true if the file exists.

Removed the **time** keyword from the POSIX Shell. The **/bin/time** is now called (which is a utility)

Changed the following messages :

```
Stopped (signal) changed to Stopped (SIGSTOP)
```

```
Stopped (tty input) changed to Stopped (SIGTTIN)
```

```
Stopped (tty output) changed to Stopped (SIGTTOU)
```

Added the **-s** option to the builtin **kill** command. The new syntax is as follows:

```
kill -s signal_name pid
kill -l
```

The **kill -l** form can be used to get the valid signal names. The following obsolescent versions are still supported:

```
kill [-signal_name] pid
kill [-signal_number] pid
```

Major Defect Fixes

See **ksh(1)** Major Defect Fixes.

Impact

The availability of executable versions of the builtin utilities creates a user awareness of the existence of such functionalities in the shell. These features

were previously documented only in the **ksh(1)** or **sh-posix(1)** entries in the HP-UX Reference. Now, separate reference pages exist pertaining to each of these new utilities mentioned in the above table.

One problem in using these utilities is that, now, a user may try to execute **fc** when using the Bourne Shell or any other shell which does not have **fc** as a builtin utility. In this case, the Bourne shell executes the executable version of **fc** from the **/bin** directory. There is no output from this **fc** because it is executed in a child sub-process. Users must be aware of the existence of these utilities. Similarly the **/bin/cd** can be used as given in the following example, but this does not effect the working directory of the caller environment.

```
find . -name file_name -exec cd
```

Alternatives and Compatibility

The POSIX Shell is essentially compatible with the HP-UX 9.0 Korn Shell. Existing users of the Korn Shell can switch between the HP-UX 9.0 Korn Shell and HP-UX 9.0 POSIX Shell with minimal impact.

Size Requirement

The size of the POSIX Shell is comparable to that of the Korn Shell. With the shared libraries linkage, the POSIX Shell executable occupies about 230 KBs of disk space.

sort(1)

Features

Digit-grouping separators are now handled in numeric sort keys specified with the **-n** option. For example, commas in the string **1,234,567** are ignored and thus the string is properly sorted based upon its arithmetic value.

Summary of Changes

The **[+pos1 [-pos2]]** form of restricted sort keys are obsolete. Using them is still supported, but the **-k keydef** option should be used instead.

Prior to release 9.0, the **field_start** and **field_end** parts of a restricted sort key incorrectly referred to fields and positions with fields starting at 0. Numbering now starts at 1.

The **-c** option no longer produces any output; the exit status is the only indication that the input was correctly presorted or not.

Performance

The sort command uses up to 16 MBs of memory to sort larger data sets if sufficient memory is available on the system. The prior limit was 1 MB.

su(1)

The **su(1)** command, when invoked without the **-** argument, removes the **HOME** and **ENV** variables from the environment before invoking a shell. This behavior is not backwards compatible with release 8.0 and was required for security reasons.

swapon(1M)

Features

The **swapon** command was modified to coordinate use of shared disk devices with the **savecore** command. Thus, for the 9.0 release, **swapon** does not enable swapping to a device which is currently in use by **savecore**. This behavior can be overridden via the new **-u** command line option.

Summary of Change

With the introduction of 2 GB memory capabilities, it is desirable from a performance perspective to allow the **savecore(1m)** command to save dump information from other than the primary swap device in parallel to the execution of the **/etc/rc** script (see the **savecore** section for more details). To ensure swapping is not inadvertently enabled on a device from which **savecore** is currently saving dump information, a communication method was established between the two commands.

The communication method is via a **lock** file. After **savecore** determines which devices contain the dump information, it saves the list of devices to the file **/etc/savecore.LCK**.

The **swapon** command was modified to check for the existence of the lockfile prior to enabling swapping on a device. This check does not affect file system swapping. If the lockfile exists, **swapon** compares each block device to which swapping is requested with the contents of the lockfile. If a match is found, **swapon** issues a warning message that indicates the specified device is in use by **savecore** and has not enabled swapping for that device. Devices for which no match is found are enabled normally. The lockfile check is performed for the following types of **swapon** commands:

```
/etc/swapon -a [-u]
```

```
/etc/swapon [-p priority][-e | -f][-u]device ...
```

A new command line option (**-u**) was added to override the lockfile check and forcibly enable swapping on devices being used by **savecore**. If **swapon** overrides the lockfile, it rewrites the contents of **/etc/savecore.LCK** omitting the devices for which swapping was enabled. This enables **savecore** to detect the case where swapping was forcibly enabled.

Machines affected

These changes only affect the HP 9000 Series 800 systems. The **-u** option is not supported on Series 700 or Series 300 systems.

Impact

Existing **/etc/rc** files continue to work without change because **savecore** defaults to invoking **swapon** after saving dump information to ensure that swapping is eventually enabled for shared devices. It is important to note that to disable swapping via the **/etc/rc** file, the command line options to **savecore** must be modified in addition to commenting out the **swapon**.

swapinfo(1M)

For file system swap areas with non-zero **RESERVE** values, the **swapinfo** command indicates the units of the **RESERVE** value are KB, but actually they are file system blocks, as given to **swapon(1M)**, which is usually 8 KBs.

sysdef(1M)

Features

This is a new command. Analyzes the named operating system and extracts the information configured into the kernel.

tail(1)

Summary of Changes

Command line execution of **tail(1)** is completely changed to the following:

```
tail [-f] [-c number | -n number | -b number] [file]
```

Please refer to the HP-UX Reference for further details. Obsolete versions of the command are not affected.

Impact

Existing scripts using the old syntax of **tail(1)** need to be modified to conform to the new one.

touch(1)

Summary of Changes

The **touch(1)** command now exits with status 0 when the **-c** option is used and the specified file does not exist. Previously, same condition would result in a non-zero exit status.

Impact

Existing scripts, relying on a successful (non-zero) exit status of `touch(1)` when `xe-c` option is used and the specified file does not exist, do not work properly.

tr(1)

Features

An additional feature to accept `ctype` and `equiv` classes with single brackets was added.

Summary of Change

An additional feature to accept `ctype` and `equiv` classes with single brackets was added.

uxgen(1M)

Kernel generation on the Series 800 now supports multiple dump devices and a gen file is allowed to leave out the specification of swap thus disabling swapping.

vi(1)

Summary of Change

The line length support (maximum number of characters per line in a file) was increased from 512 to `LINE_MAX` (2048, defined in `/usr/include/limits.h`).

volcopy(1M)

Features

This new command literally copies the file system. There are various options such as `bpi` (bits- per-inch) and size of reel that can be used while copying onto a tape.

Configuration

This command is only for HP9000 Series 800 systems.

Core Library Interfaces

Interface Name

Features

POSIX.2 conformance, increased reliability, decreased stray child processes.

Summary of Change

For POSIX.2 and XPG4 conformance, the 9.0 **system()** function no longer sets **errno** to **EINTR** and returns when it is interrupted by a signal. Instead, **system()** now continues to wait for the child process to terminate when its internal call to **waitpid()** returns **EINTR**.

Formerly, the caller of **system()** had no way to ensure that the termination status of the child process would be collected and the child thus deleted because **system()** did not indicate the process ID of the child process (preventing waiting for that specific process) and there could have been other child processes doing other things (preventing waiting for just any process to terminate). For more information on the new behavior, see the **system(3S)** entry in the HP-UX Reference.

Impact

Generally, applications that call **system()** benefit. The caller is not mistakenly resumed when the parent process handles a signal and the child process is always collected. A few applications that called **system()** might have depended on the ability to abort the **system()** function upon signal receipt even when the child process had not yet terminated. This change does not prevent the caller of **system()** from having installed signal handlers invoked during **system()** calls. The only difference is that **system()** continues to wait for the child process to terminate after the parent handles the signal.

Performance

There is no effect, except possibly a slight improvement in the time for process searches because of fewer zombie child processes.

Alternatives and Compatibility

Applications that called `system()` and depended it to return `EINTR` upon signal receipt should be changed so the child process always terminates by the time the caller of `system()` needs to resume.

Libcurses

Features

A default signal handler is added to the curses library to respond to the `SIGWINCH` signal generated as a result of any resizing operations on the window in which a curses application resides. Upon a window resize, the curses application immediately redraws the screen with the new window size. The environmental variables `LINES` and `COLUMNS` are also updated so that all the children processes of the application work with the correct window size.

In addition to the new signal handler, performance enhancements have been added to the curses library so that it is significantly faster.

Summary of Change

If there is a reduction to the window size, part of the application display is trimmed. Additionally, a snapshot of the trimmed portion is saved in the internal memory at the time of resize. This portion is not affected by the application as long as it stays invisible. If the application's cursor is trimmed, unexpected behavior results.

If a window is enlarged, any previously trimmed area is re-displayed (and re-activated). If the window is enlarged beyond its initial window size, the extra area is padded with blank spaces.

With the `SIGWINCH` support, curses applications are no longer limited to the initial window size. In other words, applications can specify coordinates beyond the initial window size at the startup time, but the extra area is not visible (and active) until the window is enlarged.

3-50 Topics Related to the HP-UX System

Impact

To use the default libcurses SIGWINCH support, there is no extra action required. A user can choose to disable SIGWINCH support by installing a local SIGWINCH handler via the `sigvector(2)` command.

Performance

With the performance improvements, the new curses library consumes fewer CPU cycles and, by employing more intelligent screen-update algorithms, often causes fewer characters to be output to the terminal. None of the library interfaces have been changed, so applications do not need to be modified in any way.

All applications using the curses library should realize some performance improvement. The extent of the improvement is dependent not only on the nature of the application, but also on the type of terminals being used. Applications that make extensive use of highlighted display characters and/or output to HP terminals benefit from the use of the new library. However, the user may notice the screen being updated differently than before.

fnmatch(3C)

The syntax is:

```
int fnmatch(const char pattern, const char *string, int flags);
```

Features

The FNM_NOESCAPE flag can be set as a part of the `flags` argument.

Summary of Change

If FNM_NOESCAPE is set in `flags`, a backslash character in `pattern` is treated as an ordinary character. If it is not set, then a backslash character in `pattern` followed by any other character matches that second character in `string`.

fprintf(3S)

Summary of Change

Two new conversion specifiers have been added: %C for wide characters (**type** `wchar_t`), and %S for wide character strings (**type** `wchar_t *`).

Impact

These additions are part of the X/Open Worldwide Portability Interface.

fscanf(3S)

Summary of Change

Two new conversion specifiers have been added: %C for wide characters (**type** `wchar_t`), and %S for wide character strings (**type** `wchar_t *`).

Impact

These additions are part of the X/Open Worldwide Portability Interface.

getgrent(3C)

Summary of Change

The maximum number of users per group was increased to 220.

glob(3C)

Summary of Change

The syntax is:

```
int glob(const char *pattern, int flags, int (*errfunc)(const
char *, int), glob_t *pglob);
```

The following items show the changes:

- Added GLOB_NOMATCH return value

- Added GLOB_ABORTED return value
- Added GLOB_NOESCAPE functionality to skip over escape in pattern
- Removed GLOB_ABEND definition
- Replaced GLOB_QUOTE with GLOB_NOESCAPE

popen(3S) and pclose(3S)

Summary of Change

Popen(3S) now invokes the POSIX command interpreter, `/bin/posix/sh` to execute the command string.

Impact

Although backward compatibility is assured in command syntax, `/bin/posix/sh` exhibits slightly different semantic behavior from `/bin/sh` (the interpreter that `popen(3S)` executed before 9.0) in the following areas:

- Trap handling
- Signal handling
- Set-user-id scripts
- Asynchronous job handling
- Different wording/spacing on some error messages
- The `/bin/posix/sh` command ignores the value of the environment variable `IFS` when executed.
- In a `chrooted` environment, `/bin/posix/sh` must be present.

A `pclose(3S)` returns a value of 127 if `/bin/posix/sh` cannot be executed. Previously, a `pclose(3S)` returned 1 for this error condition.

regerror(3C)

Summary of Change

Added REG_ESPACE as return error for out of memory message.

regexec(3C)

Summary of Change

The syntax is:

```
int regexec(const regex_t *preg, const char *string,
            size_t nmatch, regmatch_t pmatch[], int eflags);
```

The following changes were made:

- Added new members (**rm_so**, **rm_eo**) to **regmatch_t**
- Obsoleted members (**rm_sp**, **rm_ep**) in **regmatch_t**

Impact

Applications that depend on these obsoleted fields do not work any more. Also, shared library bound applications that use these structures must be recompiled.

rewind(3S)

Summary of Change

In conformance with ANSI, POSIX and X/Open standards, **rewind(3S)** no longer flushes the streams buffer.

Impact

This change may cause unexpected results in applications that expect the buffer to be flushed. Therefore, two versions of **rewind(3S)** have been provided in release 9.0. Applications that were linked with shared libraries on HP-UX 8.x releases automatically link with the old version of **rewind(3S)**. Applications that are compiled on HP-UX release 9.0 will automatically link to the new, conformant version of **rewind(3S)**.

sysconf(2)

Features

The following symbols, for which configuration values can be obtained, have been added:

- `_SC_BC_STRING_MAX`,
- `_SC_COLL_WEIGHTS_MAX`,
- `_SC_2_LOCALEDEF`,
- `_SC_SSIZE_MAX` and
- `_SC_2_FORT_RUN`.

Summary of Changes

The following symbols, which were dropped from the IEEE P1003.2 standard, have been removed:

- `SENDTO_MAX`
- `COLL_ELEM_MAX`
- `PASTE_FILES_MAX`
- `SED_PATTERN_MAX`

system(3S)

Summary of Change

A `system(3S)` now invokes the POSIX command interpreter, `/bin/posix/sh` in order to execute the command string.

Impact

Although backward compatibility is assured in command syntax, `/bin/posix/sh` exhibits slightly different semantic behavior from `/bin/sh` (the interpreter that `system(3S)` invoked before 9.0) in the following areas:

- Trap handling

- Signal handling
- Set-user-id scripts
- Asynchronous job handling
- Different wording/spacing on some error messages
- `/bin/posix/sh` ignores the value of the environment variable `IFS` when executed.
- In a `chrooted` environment, `/bin/posix/sh` must be present.

When `system(3S)` is passed a null pointer in place of a command string, it returns 1 if `/bin/posix/sh` is available, otherwise it returns 0. Previously, `system(3S)` always returned 1 in this case.

telldir(3C) and seekdir(3C)

Summary of Change

The encoding of the current directory location that `telldir()` returns was changed.

Impact

Applications that depend on the old and undocumented encoding scheme no longer work properly. Such applications should be modified so that they do not depend upon the encoding of values that `telldir()` returns. The encoding scheme was used, but never documented.

wordexp(3C)

Summary of Change

The syntax is:

```
int wordexp(const char *words, wordexp_t *pwordexp, int flags);
```

This new function performs word expansions and places the list of expanded words into the structure pointed to by `pwordexp`. The expansion would be the

same as would be performed by the POSIX shell, if the words argument is the part of a command line representing the arguments to a utility.

wordfree(3C)

Summary of Change

The syntax is:

```
void wordfree(wordexp_t *pwordexp);
```

This new function frees any memory associated with `pwordexp` from a previous call to `wordexp()`.

limits.h and definitions of FLT_MIN, DBL_MIN

Summary of Change

The constants `FLT_MIN` and `DBL_MIN` in `limits.h` are now `#defined` only when the `_XPG3` feature test macro is explicitly defined. The reason for this change is that starting with XPG4, `FLT_MIN` and `DBL_MIN` are defined in `float.h`.

Impact

In general, customers can include `limitsf.h` and `float.h`. Then, regardless of which X/Open mode they compile in, they get definitions for `FLT_MIN` and `DBL_MIN`.

Users who want to enforce XPG4 mode should define `_XOPEN_SOURCE` and `_XPG4`, and include `float.h`.

Customers who want to enforce XPG3 mode should define `_XOPEN_SOURCE` and `_XPG3`, and include `limits.h`.

Refer to `stdsyms(5)` in the HP-UX Reference for information on the `_XOPEN_SOURCE`, `_XPG3`, and `_XPG4` feature test macros.

Clusters

For the 9.0 release, Series 800 computers do NOT support clusters. A Series 800 computer cannot be a server for an HP cluster.

Communications

Communications is provided in HP-UX via several products. One or more products might be bundled with your system, and products can be purchased separately. To get more information, see products such as the following ones, which appear in Chapter 4.

- ARPA/9000
- BIND (namesaver)
- LAN/9000
- NCSNCK-RUN
- NetLS

Conformance (See Standards)

Data Communications (See Communications)

DataPair

The 10.0 release will be the last one to support DataPair.

Documentation

A Document Note for Series 800 Users

If you update a Series 800 system to HP-UX 9.0, the basic user guide is A Beginner's Guide to HP-UX (HP9000 Series 800) (Part Number B1862-90000). The Using HP-UX with HP VUE document relates only HP 9000 Series 300/400/700 systems.

A Beginner's Guide to HP-UX does not document the Posix Shell, but this does not result in any omission of functional information.

The Posix Shell is a POSIX-conforming shell similar in function to the Korn shell. The following chart shows features that may help you decide which shell will work better for the kind of work you do.

Comparison of Shell Features

Features	Description	Bourne	Posix/Korn/Key	C
Command history	A feature allowing commands to be stored in a buffer, then modified and reused.	No	Yes	Yes
Line editing	The ability to modify the current or previous command lines with a text editor.	No	Yes	No
File name completion	The ability to automatically finish typing file names	No	Yes	Yes

in command lines.

Alias command	A feature allowing users to rename commands, automatically include command options, or abbreviate long command lines.	No	Yes	Yes
------------------	--	----	-----	-----

Restricted shells	A security feature providing a controlled environment with limited capabilities.	Yes	Yes*	No
----------------------	---	-----	------	----

* The POSIX shell does not have a restricted shell.
See the Shells User's Guide for information.

Job control	Tools for tracking and accessing processes that run in the background. See Shells: User's Guide.	No	Yes	Yes
----------------	---	----	-----	-----

New Documents

DTC Device File Access Utilities (DDFA), Part Number B1014-90012

This manual is part of the System Administration Documentation Set and contains information on using the DTC Device File Access Utilities and Telnet Port Identification features of ARPA/9000. Refer to HP ARPA SERVICES later in these release notes.

Online Help

If you use HP VUE or the X Window System, you can use the HP Help System. HP Help provides a wealth of help information that you can browse, search, display, and print. This information displays in a window called the Help Manager.

To start the Help Manager, click on the books and question mark icon on the front panel.

3-60 Topics Related to the HP-UX System

Help Manager includes the following topics (it might also include additional help, depending on what optional products you have purchased):

- The HP Visual User Environment Help, Version 3.0 document provides information on HP VUE. Refer to this help to learn about HP VUE and its features.
- The HP-UX 9.0 Operating System Help document provides information on using the HP-UX operating system. Refer to this help to perform common HP-UX tasks, determine the cause of HP-UX errors, and locate information about HP-UX manuals.
- If you have a pre-loaded operating system (Instant Ignition), Welcome Messages provides an introduction to your new workstation and the HP VUE interface. It also explains how to create user accounts and change passwords.

You can also get context-sensitive help by doing either of the following:

- Move the cursor to the object or icon for which you want help, then press [F1], the **help** key.
- Choose the up arrow above the Help control (the books and question mark) in the Front Panel. The Help subpanel appears. Choose {On Item} in the Help subpanel.

The cursor changes to a question mark (?). Choose the object or icon for which you want help.

For more information about online help, refer to Chapter 3 of Using HP-UX.

Documentation Errata

acctcom(1M) Command

There is an error in the acctcom(1M) entry in the HP-UX Reference. In the acctcom(1M) entry, the line:

The command name is preceded by a # if it was executed with super-user privileges.

is incorrect. It should read:

The command name is preceded by a # if super-user privileges

were required to execute the command.

For example, a user can be logged in as root and executes `date` to check the time. Since this command does not require super-user privilege, it is shown by `acctcom` without the `#` character on the line. On the other hand, executing `date 0731180092` to set the time does require super-user privileges. So, it is shown by `acctcom` with the `#` character on the line.

In the bugs section of the entry in the HP-UX Reference, add the following information:

```
The accounting flag is not cleared when one processes execs
another, but only when one process forks another. One side-effect
of this is that some processes are marked with #, when
users do not expect this.
```

For example, the `login` program uses super-user privileges to assume the identity of the user who is logging-in, setting the ASU bit (which ultimately causes the `#` character in the `acctcom` output). After assuming the user's identity, it execs the user's shell. Since the `exec` does not clear the ASU flag, the shell inherits it and the flag is marked with a `#` in the `acctcom` output.

Drivers

The information in this section relates to drivers, which themselves relate to the kernel and, perhaps, to other products. Therefore, the information in this section might be duplicated elsewhere.

LAN Driver

The LAN Driver, `LAN3`, controls the LAN portion of the LAN/Console card as well as the LAN only card. It provides IEEE802.3/ETHERNET service and the card diagnostic functions for the Series 8x7 and 890 HP-UX systems.

`LAN3` was originally designed to support the LAN only card on Series 8x7 HP-UX systems for the 8.02 release. It was modified to synchronize with Terminal I/O code (`Console`) to support the LAN/Console card on Series 8x7 and 890 HP-UX systems for the 9.0 release. The functionality of `LAN3` for the

9.0 release is the same as for the 8.02 release. Performance of **LAN3** for the 9.0 release has improved.

Filesets (See Partitions)

Files and File Systems

The following sections indicate changes in files or file systems.

/etc/group

Membership in each group is limited to 200 users. A workaround is possible if a system administrator creates multiple entries for a group. SAM supports this workaround if the entries are adjacent and the first three fields that define the group are identical.

lseek(2)

Features

Any **lseek(2)** operations on block or character devices greater than 2 GBs in size might legally take a negative offset as a parameter and return negative offsets. Returning a -1 is always an error for **lseek(2)**.

Summary of Change

Prior to 9.0, HP-UX did not allow file systems larger than 2 GBs. The reason was that the kernel and the system-call interface (**lseek**) used the type **off_t** for byte offsets, and **off_t** was a signed 32-bit integer. In 9.0, a maximum size of 4 GBs for file systems was been implemented by changing the meaning of **lseek**'s offset to be unsigned for character and block devices only. On a regular file, **lseek** is not changed, and the maximum size of regular files is still 2 GBs. The change to **lseek** does not violate standards because it affects only special files whose behavior is left undefined by the standards. A -1 is always an error

return for `lseek(2)`, even when encountered on disk sections greater than 2 GBs.

Impact

Some users might check an error return from `lseek(2)` with the following code:

```
if ( lseek(fd, offset, whence) < 0 )
    /* error */
```

Doing this incorrectly reports an error if for a seek beyond 2 GBs on a character special file or on a block greater than 2 GBs. Such code should be changed to read:

```
if ( lseek(fd, offset, whence) == -1 )
    /* error */
```

Only block and character special files corresponding to devices that are larger than 2 GBs are affected.

Performance

The change in `lseek(2)` have no effect on performance.

Alternatives and Compatibility

See the section above named **Impact**.

Configurations Affected

- 2-way striped HP 22520 disks using section 2.
- Two HP 22520 disks that have section 2 mirrored using the DataPair mirror driver.
- LVM, using any combination of disks up to 4 GBs.

Maximum File System Size

Features

In 9.0, the maximum file system size was increased from 2 GBs to 4 GBs. This increase affects only the size of an entire file system, not the size of individual files, which still have a maximum size of 2 GBs.

Impact

See the section describing changes to `lseek(2)`.

Performance

This change has no performance effect.

Configurations Supported

File systems of greater than 2 GBs can be created on:

- 2-way striped HP 22520 disks using section 2.
- Two HP 22520 mirrored disks using the DataPair mirror driver.
- LVM using any combination of disks desired up to 4 GBs. All LVM features are supported.

Internationalization

Localized User Interface Information Series 800

HP-UX commands, CUE (Character-terminal User Environment), TSM (Terminal Session Manager), and HP VUE (Visual User Environment) have the localized user interface.

- Message catalogs of HP-UX commands are localized into 6 languages (provided in LANG_TYPE-CAT fileset):

French, German, Japanese, Korean,
Simplified Chinese, Traditional Chinese

- CUE message catalog is localized into 3 languages (provided in LANG_TYPE-CAT fileset):

French, German, Japanese

- TSM menus and messages are localized into 2 languages (provided in LANG_TYPE-CAT fileset):

French and German

- Message catalogs and environment files of HP VUE are localized into 4 languages (provided in LANG_TYPE-X11 fileset):

Japanese, Korean, Simplified Chinese, Traditional Chinese

- Selected HP-UX Reference entries in Section 1 (about 50 files) are localized into Japanese (provided in JAPANESE-MAN fileset)

Since all localized files are provided in either on CD or on UEE (User Enhanced Environment) tape, which is shipped with the HP-UX Runtime tapes, customers can selectively install the localized filesets for their native language. The separate NLIO product is also required to enable the localized user interface in Asian languages.

The localized user interface in Japanese requires about 1.1 MBs of disk space, other Asian languages (Korean, Simplified Chinese, and Traditional Chinese) require about 0.6 MBs per language, and European languages (French and German) require about 0.4 MBs per language.

Using the Bourne Shell as the Default

In 9.0, the PATH variable is specified so the system finds the Posix Shell before it finds the Bourne Shell. People who want the Bourne Shell should move `/bin/sh` ahead of `/bin/posix/sh` in the PATH specification in their `/etc/.profile` and `/etc/csh.login` files.

Regardless of the specification of PATH, programmatic access to the shell through `popen(3C)` and `system(3C)` uses the internationalized Bourne Shell when an Asian multi-byte language is used.

Kernel

The following, rather long, section describes what is new, changed, and removed from the kernel.

HPUXBOOT

Features

For 9.0, the old HPUXBOOT kernel loader was completely rewritten. The new loader attempts to be functionally and visibly equivalent to the old HPUXBOOT. However, some differences exist.

Summary of Change

The new kernel loader for release 9.0 is a standalone application using none of the kernel source as did the old HPUXBOOT. It depends only on a few kernel `include` files that are typically exported to user commands.

New Kernel Interface

For release 9.0, the kernel interface was adjusted to accept an `argc`, `argv`, `env` set of arguments. This is the type of interface the new kernel loader uses when launching any type of application (not necessarily a kernel). Any release 9.0 kernel was modified to accept both the old HPUXBOOT interface and the new kernel interface. This interface is more flexible than the 8.0 kernel interface.

The new kernel interface is incompatible with pre-9.0 kernels. The older kernels will probably not boot. This circumstance should only exist as the result of a failed installation or update. If this happens, reinstalling with the support tape should fix the problem. To be absolutely sure that a system can be recovered, a backup of an 8.0 boot LIF should be kept.

Cross booting

One disadvantage of new kernel interface is that it does not have the extensive code required to configure all computer hardware. This makes the ability to boot to ISL from a device on one IO adapter and boot the kernel from a different device on a different IO adapter, impractical. That means, should you choose to boot to ISL from a device, you can only use new kernel

interface to boot a kernel from a device attached to the same device adapter (SCSI/HP-IB/FL/etc). For example, it is not possible to boot ISL from tape at HP-PB address 44.2 and have new kernel interface go through HP-PB address 52.3 to access a disk and get a kernel there. Users with CIO systems are not affected unless they require booting from an adapter on a different CIO channel or bus converter. And in that case, the same restrictions apply.

As this function is used by the support tape to fix boot areas, provisions are now being made to solve the problem without need for this function. It should be a menu selectable item on the support tape. The removal of this function does not affect an installation.

Booting from tape

Due to firmware limitations, new kernel interface does not support the file-skip feature on tape. This feature was used by both the install and support tapes. The install and support tape kernels were placed behind a boot LIF on the tape. To boot, a skip count of 1 was given to **HPUXBOOT** which then skipped the first file, the LIF, to find and load the kernel. Replacing this feature is the ability to boot directly from LIF. Both the install and support tape kernels are placed directly in the boot LIF.

Since the kernels are placed directly in the LIF, they might be on tape at a higher block size. This reduces wasted tape due to inter-record gaps and reduces the number of tape accesses necessary to load the kernel. Fewer accesses means a faster boot time.

The visual impact is the removal of the **skip** field in the boot command and the introduction of **:INSTALL** or **:RECOVERY** as the target file name. For example, booting an install tape for release 8.0 required the command

```
ISL> hpux (;0xa0000,1)
```

The new command is:

```
ISL> hpux (;0xa0000):INSTALL
```

This feature also allows a simplified version of booting the install and support tapes.

Large Capacity Disks

A restriction, which would have existed for HPUXBOOT, also exists for new kernel interface. When dealing with disk drives that have a capacity greater than 2 GBs, only partitions that exist completely in the first 2 GBs of the drive are available as root partitions. The new kernel interface cannot, due to architectural and firmware restrictions, address data beyond 2 GBs of a disk or tape. LVM disks do not pose a problem as root never exists beyond 2 GBs of the beginning of the media.

LIF boot

The new new kernel interface supports booting a kernel directly from a LIF. For 9.0, this feature is intended only for booting the Install and Support tapes.

LAN boot

As diskless is not supported for 9.0, the new kernel interface only supports a LAN boot internally.

Logical Volume Manager

The new kernel interface supports LVM. To boot an LVM disk, the command is:

```
ISL> hpux (;0)/hp-ux
```

This assumes you choose to boot `/hp-ux` and not some other kernel. Regarding LVM, two new options are maintenance mode, `-lm`, and quorum mode, `-lq`. See the System Administration Tasks document for more information.

B1 File Systems

Whereas HPUXBOOT needed to be built specifically to run on B1 systems, the new kernel interface does not. Any new kernel interface supports booting from a B1 file system.

Build Procedure

The build procedure for the new kernel interface is far simpler than that for HPUXBOOT; the procedure is as simple as running `make`. The entire build process takes less than 30 minutes. There is an option to build new kernel interface

that runs as a simulation and one that runs on any series 800 computer. Running as a simulation helps you find and fix new kernel interface defects.

Performance

The new kernel interface should reduce the time it takes to launch a kernel. Elimination of redundant hardware configuration, smaller size, and better algorithms have yielded boot time improvement of as much as 50% (data collected on an HP 9000 Model 870 four-way MP system, reflecting time from ISL to kernel launch).

Size

The new kernel interface is much smaller than **HPUXBOOT**. On a disk, release 8.0 **HPUXBOOT** occupied 1.4 MBs of a 2 MB maximum LIF. Other LIF applications brought the size to very near the limit. The new kernel interface occupies only 200 KBs on disk. With other LIF applications, this amounts to a LIF that occupies less than 1 MB. Since a backup LIF exists in **/usr/lib**, this means that there is over 1 MB of disk space savings for the user.

In memory there is also an advantage. **HPUXBOOT** occupied just under 4 MBs of memory. This restricted the kernel loadable image size to 4 MBs. The new kernel interface occupies just over 1 MB of memory. This means the new kernel interface was allowed to move much higher in memory, which freed space for the kernel loadable image. Now, the new kernel interface can load a kernel image of up to 6 MBs.

Obsolete Kernel Features

Features

Changes to the kernel's feature set are generally backwards compatible. That is, they are pure supersets to previous releases. This section discusses the exceptions.

The following features have been declared obsolete as of the 9.0 release:

- Diskless server/client support.
- Appletalk. (Appletalk is a trademark of Apple Computer, Inc.)
- Graphics device support.

3-70 Topics Related to the HP-UX System

- MAP products support.
- The X.25 product is now an external product. There have been several kernel generation parameter changes. Finally, the logical page size (as opposed to the physical page size) was changed.

Kernel Support for Diskless Server and Clients

Summary of Change

The 9.0 Series 800 kernel no longer handles diskless nodes as either clients or servers in any type of network. It is no longer possible to generate a kernel with diskless support.

Impact

Existing Context Dependent Files (CDF's) are still readable, but every Series 800 should have its own root disk. Other file systems can be NFS mounted.

The diskless driver, `rdu`, was declared obsolete, and a warning to that effect is given by `uxgen` for any kernel generation files that have it included.

Alternatives

Heterogeneous clusters either have to have a Series 300 or 700 identified as the server. Another possibility is to use a root device for each system and use NFS to mount the rest of the files from the server. A final alternative is to continue to use an 8.x release, which continues to be supported for this purpose.

Computers Affected

All Series 800 computers that are either diskless servers or clients.

Kernel Support and Series 800 Graphics

Summary of Change

The 9.0 Series 800 kernel no longer supports the `graph0`, `graph2`, `display0`, `hil`, `hilkbd`, or `ite` drivers. For more information, see the Starbase topic.

Impact

Series 800 computers such as the Model 845 SRX, which have special graphics devices (ie, devices which require a special card in the computer), are no longer supported. Bit mapped monitors do not work either as system consoles or as terminals. HIL devices such as keyboards, mice, and ID modules do not work. Graphics devices on normal IO cards do work. This includes anything on a RS-232, HP-IB, or Centronics bus such as plotters, printers, graphics terminals. X-terminals connected via LAN still work.

Old kernel generation files that have the graphics drivers configured cause `uxgen` to give a warning.

Alternatives

Replace bit mapped monitors with X terminals and use an inexpensive terminal for the system console. Another possibility is to replace the Series 800 with a Series 700 workstation.

Computers Affected

All Series 800 computers with Appletalk interfaces.

MAP Products

Summary of Change

The following products have been obsoleted and will not be supported by the 9.0 kernel:

- OSI Express Carrierband (36969A)
- OSI Express Broadband (36970A)
- HP MAP 3.0 MMS for Series 800 (B1007A)
- HP MAP 3.0 FTAM for Series 800 (B1008A)

Impact

The following statements result in a warning that the options requested are ignored:

- `include osiulipc0;`
- `include osi0;`
- `include ositest;`
- `include netdiag0;`
- `include ktltest;`

The 9.0 kernel has no drivers capable of talking to the obsolete `osi4` card.

Computers Affected

All Series 800 computers with the `osi4` card. (Any computer with the MAP fileset.)

X.25

Summary of Change

The X.25 product is external as of the 9.0 release. This means that the X.25 bits may not be on the same product tape as the kernel.

Impact

Updating a system that is already configured for X.25 results in a working kernel that cannot talk to the X.25 hardware until the X.25 product is installed.

Computers Affected

All 800 series computers with X.25 interfaces.

Logical Page Size

Summary of Change

The kernel now handles the memory of all Series 800 computers in pages of 4 KBs (4096 bytes). This is a change from releases 8.0, 8.06, and 8.06+, where the page size was 2 KBs (2048).

Impact

Applications, device drivers, and commands that deal with page aligned data that was supported at 8.0, 8.06 or 8.06+ might need to be recompiled or re-coded, depending on why the code needs to know about page alignments and whether a dynamic test, system header file variable (NBPG), or hard coded value was used.

Also, tunable kernel parameters that are based on numbers of pages (or page size) are twice as large. Specifically, the default value for the parameter **iomemsize** is twice as large, so any user-configured value for **bufpages** results in more space being used for the buffer cache. See the section Buffer Cache under Memory for discussions of the new algorithms and the affects of the kernel parameters.

Computers Affected

All Series 800 computers.

Kernel Parameters Changes

fs_async Parameter

Summary of Change

The default value of the kernel parameter **fs_async** is 0 in HP-UX 9.0. This parameter was present in HP-UX since Release 8.0. The default value was 0 in every release except 8.06, where it was 1.

Impact

Administrators who update from 8.06 to 9.0, and who do not assign an explicit value to **fs_async** when generating their kernel, find that the value has changed from 1 to 0. Other administrators see no change.

Performance

The setting of **fs_async** gives the System Administrator some control over the number of file system writes that are synchronous. The possible values are 0 and 1. With **fs_async** set to 0, more file system structures are written synchronously than with it set to 1. The advantage of the 0 setting is that,

after a crash, **fsck** is more likely to recover the disk automatically, and with minimal loss of data. The disadvantage of 0 is lessened performance.

Alternatives and Compatibility

System Administrators can always override the default when generating a new kernel.

maxdsiz, maxtsiz, and maxssiz Parameters

The **maxdsiz** (max data size), **maxtsiz** (max text size) and **maxssiz** (max stack size) are specified in bytes on the Series 800 at release 9.0. Specifying these parameters in bytes is consistent with other HP platforms (Series 300, Series 700) and is consistent with the OSF specification. In previous releases of HP-UX on Series 800 computers, these parameters were specified as the number of pages (e.g. **maxdsiz** 32768).

Summary of Change

The **maxdsiz**, **maxtsiz**, **maxssiz** parameters are now specified in bytes.

Impact

To use the default values for these parameters, no action is required. The user should not notice any change.

If the user changes these parameters and now forgets to change to the new format, repeated system failures might occur. The user may get a message from **init** at boot time such as **init: out of memory**. Also, at boot time, the **fsck** command may report **unable to read blk xxx**. The messages might cause a user to think something is wrong with the file system. Both messages are the result of **max*siz** specifying pages instead of bytes.

Performance

There is no impact on the performance of the kernel by this change.

Alternatives and Compatibility

The KERN-BLD customization script checks for any specification of **maxdsiz**, **maxtsiz** or **maxssiz**. If specified, the script uses the default values and issue an appropriate message to the user. The user must then regenerate the kernel

after adjusting the values. (The kernel booted by update does use the system default values.) The values are not automatically changed because there is a set of values that overlap such that it would be impossible to determine if the user meant bytes or pages.

By using the default values, the user gets a kernel that works. By reviewing the update log, users discover they need to take action because they have changed these particular parameters. However, if a user updates the KERN-BLD fileset after correcting parameter values, and the gen file is syntactically incorrect, the update log claims the parameter values are not appropriate.

A second script is supplied in `/etc/newconfig/fix2_90_gen` that can be used to convert any other `gen` files. The defaults are large and only users that are running large applications or simulations would need to change them.

n`text` Parameter

Beginning with the 8.0 release, the `ntext` parameter, which was described in the `uxgen(1M)` entry in the HP-UX Reference, is no longer used by the kernel. It is no longer a tunable parameter.

bufpages Parameter

The meaning of the parameter `bufpages` has not actually changed. It still refers to the number of pages in the file system buffer cache. But the page size has doubled (gone from 2048 bytes to 4096 bytes), so the kernel-generation files that set `bufpages` may set it too high for 9.0.

Languages (Computer)

With the 9.0 release, many languages are available as separate products. Such products have their own release notes. The release notes for them are placed on the system during the update process. After the update, they appear in the following directory:

`/etc/newconfig/90RelNotes`

For example, this situation applies to the C, Pascal, and FORTRAN languages.

Some language products include a symbolic debugger, which is placed on the system during the update process. For more information, see the topic named Symbolic Debugger.

Localization (See Internationalization)

Licenses (Product)

If your product has multiple licenses, an installation might be performed for the most restrictive license. Your HP representative has information about this.

Linker

New Features

The following items describe the linker options added to HP-UX 9.0 since the 8.07 release. To get more information, see the `ld(1)` entry in the HP-UX Reference and the 9.0 Programming on HP-UX manual.

<code>-a archive_shared</code>	Indicates that within each directory searched at link-time to find a library specified with <code>-l</code> , the archive form of the library is preferred, but the shared form is allowed.
--------------------------------	---

-a shared_archive	Indicates that within each directory searched at link-time to find a library specified with -l, the shared form of the library is preferred, but the archive form is allowed.
-l: specify library	Similar to -l except not affected by setting of -a option.
-B restricted	Restrict symbols available to shared libraries seen at link time to those present when program first starts up.
+s	Indicate that the SHLIB_PATH environment variable may be used by the shared library loader to locate shared libraries at run time.
+b path-list	Specify a list of directories that are to be searched at run time by the shared library loader to locate the shared libraries needed by the program.
+FP flags	Specify the initial trap enable bits for floating point operations.
+df name	Feedback Directed Positioning (FDP) specify profile database
+pgm name	Feedback Directed Positioning (FDP) specify profile database lookup name

Additionally, users who move from HP-UX 8.0 directly to HP-UX 9.0 see these new linker options.

+e	List symbols to be exported from a shared library
-h	List symbols not to be exported from a shared library
+I	Wspecify an entry point for a shared library to be called when the library is loaded.
-c	List an indirect textfile that contains linker options
-B non-fatal	Indicate that unresolved symbols in shared libraries are not to be considered fatal
-I	Feedback Directed Positioning (FDP) reposition procedures
-P	Feedback Directed Positioning (FDP) reposition procedures
-O	Turn on linker optimizations

3-78 Topics Related to the HP-UX System

LP Spooler

Removed Features

The `/etc/mklp` script, which configures printers and the LP Spooler, and its associated HP-UX Reference page was removed. HP recommends that people use SAM to configure printers and the Spooler.

HP-UX Math Libraries

HP-UX supplies the following math libraries:

<code>libm</code> and <code>libM</code>	The C math libraries.
<code>libcl</code>	The math functions within the FORTRAN and Pascal compiler library.
<code>libvec</code>	The vector library.

The HP-UX Series 700/800 systems include two versions of each math library.

- The first version relates to the PA-RISC 1.0 architecture (PA1.0). Changes to this library relate to repaired features.
- The second version relates to the PA-RISC 1.1 architecture (PA1.1). Changes to this library relate to repaired features and improved performance, increased precision, and new features. The PA1.1 versions of the libraries have the following characteristics.
 - Faster than the PA1.0 versions for most applications.
 - More precise than the PA1.0 versions for most applications.
 - Contain new functions and features in addition to those specified by the POSIX, SVID, OSF AES, and XPG3 standards.

The PA1.0 versions execute on any Series 700/800 system.

The PA1.1 versions execute on all Series 700 systems and on Series 8x7 systems (Series 800 systems with model numbers whose last digit is 7). Applications

that require maximum performance or that can benefit from some of the new functions available in the PA1.1 versions should use these libraries.

Because the PA1.1 versions use more sophisticated algorithms, they yield more precise results than those yielded by the PA1.0 versions.

HP offers the following recommendations:

- If an application must run on all Series 800 systems, including those based on PA1.0, use the PA1.0 versions of the libraries.
- If reproducing results in an application from release to release is more important than the performance of the application, use the PA1.0 versions of the libraries.
- If an application requires maximum performance combined with maximum precision, use the PA1.1 versions of the libraries.

For complete information about HP-UX math libraries, see the HP-UX Floating-Point Guide.

Selecting Different Versions of the Math Libraries

If you use a compilation command (`f77`, `cc`, `CC`, `pc`) to execute the link editor, `ld`, the selection of the PA1.0 or PA1.1 math libraries is driven by the `+DA` compiler option. This option lets you to generate either PA1.0 code (`+DA1.0`) or PA1.1 code (`+DA1.1`).

When you select `+DA1.0`, the compilation command invokes `ld` with a library search path of `/lib` and `/usr/lib`. The PA1.0 libraries are stored in these directories.

When you select `+DA1.1`, the compilation command invokes `ld` with the directories named `/lib/pa1.1` and `/usr/lib/pa1.1` at the head of the library search path. The PA1.1 libraries are stored in these directories.

On all Series 800 computers, the default value of the `+DA` option is `+DA1.0`, because the primary concern for most Series 800 applications is compatibility with previous versions of the application, and because not all Series 800 models support PA1.1.

On all Series 700 machines, the default value of the **+DA** option is **+DA1.1**, because the primary concern for most Series 700 applications is maximum performance, and because all Series 700 models support PA1.1.

You may use any of the following methods to override the default library search path set by the **+DA** option:

- Specify the **+DA** option explicitly on the command line.
- Specify a library directory by using the compiler command-line option **-L** (for C) or **-Wl, -L** (for Fortran and Pascal).
- Link in the library directly by specifying the complete path name on the command line.

See **Programming on HP-UX** for information on the last two linking methods. The **ld** command does not recognize the **+DA** option, so if you invoke **ld** explicitly (instead of using a compiler command), you must use one of these two methods.

There are some restrictions on your use of the **+DA** option:

- Applications built with **+DA1.0** that use archive libraries run on all Series 700 and Series 800 systems. However, applications built with **+DA1.1** that use archive libraries run on Series 700 systems and Series 8x7 systems.
- Applications that use shared libraries and that are built with **+DA1.1** do not run on a Series 800 system that is running a pre-9.0 version of HP-UX.
- The PA1.1 versions of the math libraries are available only on systems running HP-UX Release 9.0.

If these restrictions are taken into account, an application built on one system (Series 700 or 800) runs on the other series's systems and yield identical results, regardless of which version of the library is selected. This happens because the shared versions of the libraries are identically positioned in the system directories on Series 700 and Series 800 systems and because of the permanent nature of archive library linking.

Locations of the Math Libraries

The library directories (`/lib` or `/usr/lib`) each have a subdirectory called `pa1.1`. The PA1.1 versions of the libraries are in this subdirectory, while the PA1.0 versions are in the parent directory (`/lib` or `/usr/lib`). The complete list of math and vector library path names follows; PA1.1 libraries first, PA1.0 libraries second, and profiling libraries third. For performance reasons, HP provides the PA1.1 versions of `libm` and `libM` only as archive libraries. This can affect an application if C math functions are used in a shared library that is `shl_loaded` by another shared library (which we'll call the **parent** library), and if the math functions are not used in the parent library. In this case, an application linked on HP-UX 9.0 with the parent library fails at runtime due to unresolved external references to the math functions if and when the other library is `shl_loaded`.

The PA1.1 library path names follow:

<code>/lib/pa1.1/libm.a</code>	C math library, SVID, archive version
<code>/lib/pa1.1/libM.a</code>	C math library, POSIX, archive version
<code>/usr/lib/pa1.1/libcl.a</code>	Compiler library (includes FORTRAN math library), archive version
<code>/usr/lib/pa1.1/libcl.sl</code>	Compiler library (includes FORTRAN math library), shared version
<code>/usr/lib/pa1.1/libvec.a</code>	Vector library, archive version (this version was in <code>/usr/lib/libvec.a</code> at Release 8.0).

The PA1.0 library path names follow:

<code>/lib/libm.a</code>	C math library, SVID, archive version
<code>/lib/libm.sl</code>	C math library, SVID, shared version
<code>/lib/libM.a</code>	C math library, POSIX, archive version
<code>/lib/libM.sl</code>	C math library, POSIX, shared version
<code>/usr/lib/libcl.a</code>	Compiler library (includes FORTRAN math library), archive version
<code>/usr/lib/libcl.sl</code>	Compiler library (includes FORTRAN math library), shared version

`/usr/lib/libvec.a` Vector library, archive version (this version is new at Release 9.0; for Release 8.0, this path name contained the PA1.1 version)

The profiling library path names follow:

`/lib/libp/libm.a` C math library, SVID, profiling version, archive version, PA1.0

`/lib/libp/libM.a` C math library, POSIX, profiling version, archive version, PA1.0

`/usr/lib/pa1.1/libp/libvec.a` Vector library, profiling version, archive version, PA1.1

Contents of the PA1.1 Math Libraries

In addition to superior performance and precision, the PA1.1 scalar math libraries provide routines that are not available in the PA1.0 libraries. The `libm`, `libM`, and `libcl` libraries all support the `fpgetround` suite of routines, which enable you to manipulate the floating-point control register. The `libm` and `libM` libraries also support the following functionality:

- Float versions of many mathematical functions
- Degree-valued trigonometric functions
- A group of functions recommended by the IEEE standard
- A group of BSD4.3 functions

These functions are defined in `math.h`. However, the functions work outside the POSIX, SVID, and ANSI C standards. If you compile in ANSI mode (with the `-Aa` option), only the declarations of functions specified by the ANSI C standard are ordinarily visible to your program. Therefore, to make the new functions visible, take one of the following actions:

- If your program does not use ANSI C features, compile in compatibility mode (with the default `-Ac` option), which makes both ANSI and non-ANSI function declarations visible:

```
cc program_name.c -lm
```

- Compile with **-Aa** and use the **-D** option to define the macro **_HPUX_SOURCE** on the command line. If you define this macro, all the function declarations are visible. If you compile with **-Ac**, this macro is defined automatically.

```
cc -Aa -D_HPUX_SOURCE program_name.c -lM
```

If you use the float-type math functions, only the second of the above methods is available.

The above functions are implemented only in the PA1.1 versions of **libm** and **libM**, although they are defined in **math.h**, which applies to both the PA1.0 and PA1.1 versions. Therefore, if you use any of these functions in your application and inadvertently link with the PA1.0 version of the library, the compilation succeeds, but the linker fails with unresolved references to the function names. If you compile on a Series 700, you link with the PA1.1 library by default. If you compile on a Series 800, specify **+DA1.1** to link with the PA1.1 library.

The functions in the **fpgetround** suite follow:

- **fpgetround()**, **fpsetround()**
- **fpgetmask()**, **fpsetmask()**
- **fpgetsticky()**, **fpsetsticky()**
- **fpgetfastmode()**, **fpsetfastmode()**
- **fpsetdefaults()**
- **fpgetcontrol()**, **fpsetcontrol()**

The degree-valued trigonometric functions in the C library follow:

- **cosd()**, **sind()**, **tand()**
- **acosd()**, **asind()**, **atand()**, **atan2d()**

The float type math functions in the C library follow:

- **cosf()**, **sinf()**, **tanf()**, **cosdf()**, **sindf()**, **tandf()**
- **acosf()**, **asinf()**, **atanf()**, **atan2f()**, **acosdf()**, **asindf()**, **atandf()**, **atan2df()**
- **coshf()**, **sinhf()**, **tanhf()**

- `expf()`, `fabsf()`, `fmodf()`, `logf()`, `log10f()`, `log2f()`, `powf()`, `sqrtf()`, `cbrtf()`, `copysignf()`
- `isinff()`, `isnanf()`, `finitef()`, `fpclassifyf()`

The IEEE recommended functions in the C library are `copysign()`, `scalb()`, `logb()`, `finite()`, and `fpclassify()` (`fpclassify()` is the HP version of the `class()` function recommended by the standard).

The BSD4.3 functions in the C library are `acosh()`, `asinh()`, `atanh()`, `cbrt()`, `rint()`, and `drem()`.

The C library also supports the nonstandard function `log2()`.

For details on all of these functions and on how to use them, refer to the online reference pages or to the **HP-UX Reference**.

NOTE: The lint library `/usr/lib/llib-lm` was updated to include the new PA1.1 functions.

Logical Volume Manager (LVM)

All new pre-installed systems are shipped configured with Logical Volumes instead of HP-UX hard partitions. New software installations of HP-UX 9.0 default to Logical Volumes. Update customers are strongly encouraged to migrate from HP-UX hard partitions to Logical Volumes. HP recommends that you gain a full understanding of LVM Concepts by reading the LVM chapter in the *How HP-UX Works: Concepts for the System Administrator* document before you migrate to Logical Volumes.

The Logical Volume Manager is a new disk management subsystem for HP-UX 9.0. Ported from OSF1.0, and enhanced by HP, LVM allows users to handle physical disks (Physical Volumes) as Logical Volumes. Logical Volumes in turn allow the creation of filesystems that can span multiple Physical Volumes. A Logical Volume can be used directly in place of a raw disk partition. Since LVM is a subsystem between the physical disk driver and the filesystem, a Logical Volume can be used in place of a Physical Volume in any HP-UX command such as `mount`, `bdf`, or `dd`. However, since the size of a Logical Volume is not static, the user has greater flexibility in configuration of filesystems and the utilization of physical volumes.

The base Logical Volume Manager is bundled with the HP-UX 9.0 operating system. LVM mirroring can be added via the product named LVM-MIRRORING. This product contains all of the LVM mirroring features. The base LVM product does not contain the mirroring capability.

Features

- **Dynamic Logical Volumes:** A LV is equivalent to a traditional disk or disk partition but can be extended as needed.
- **Logical Volumes can span Physical Volumes:** A Logical Volume can span up to 255 Physical Volumes. This allows the size of a LV to become as large as 4 GBs, which is an HP-UX system limitation.
- **Support of HP-FL, HP-IB, or SCSI Physical Volumes.**
- **Higher availability through 2 or 3 way mirroring:** This features is supported for any logical volume including a root filesystem. In addition, mirrored Logical Volumes can be split to allow online backup of a split mirror. The logical volume can then be remerged and synced with the original. Mirroring is supported for HP-FL and SCSI disks only. These features can be obtained via the DATAPAIR PLUS product.
- **Bad Block Relocation:** LVM attempts to transparently recover media errors. Bad block relocation is supported only on HP-FL or SCSI Physical Volumes, and is not supported on the Root Logical Volume.
- **Root, Swap and Dump Logical Volumes.** This allows the user to boot from a filesystem residing on a possibly mirrored logical volume. The root, swap, and dump logical volumes must be contiguous; meaning they can not span multiple physical volumes.
- **Multiple Dump Devices:** HP-UX 9.0 supports multiple dump locations. LVM also supports this feature in that multiple logical volumes (up to 14) can be configured as dump volumes. A dump logical volume must be contiguous.
- **Filesystem Extensibility:** An unmounted **hfs** filesystem can be increased dynamically to utilize an unused portion of a existing Logical Volume or to extend onto a newly added Physical Volume.

- Coexistence and support of existing products from HP including raw databases, NFS, Switchover/UX and DataPair. HP recommends that users convert existing DataPair filesystems to LVM.
- Use of the System Administration Tool `/usr/bin/sam` for configuring LVM and aiding in migrating to LVM.

LVM System Installation

Pre-Installed 9.0 Systems

Factory installed 9.0 systems are preconfigured with a LVM root Logical Volume mounted on `/` and other filesystems such as `/usr` and `/tmp` in the Root Volume Group.

The customer can then use LVM commands directly or use `/usr/bin/sam` to add additional disks or create additional filesystems with LVM.

Migrating Root Disk to LVM

The *Installing and Updating HP-UX 9.0* document has a complete description of the planning, tools, and installation process required to migrate the root disk to a logical volume.

Migrating Non-Root Disks to LVM

The *Installing and Updating HP-UX 9.0* document has a complete description of the planning, and update process required to move from release 8.0x to 9.0. This same document has information about migrating non-root disks and points a user to the *System Administration Tasks* document, which describes the process in detail.

In regard to migration, be aware that some applications might require modification, after the migration, to run on an LVM system. Routines that use device files need to be examined. For example, using `devnm` to find a device file prints `/dev/vg00`, and some commands still use special device files in the form of `/dev/dsk/xxx`. Also, do not rely on `/etc/disktab` to provide sizes of logical volumes.

OSF Functionality Details

The LVM implements a logical disk. Application software can interface with logical disks through either the character or block device interface just like the character or block device interface provided by physical disks. The programming interface to these logical disks for ordinary applications is identical to any other character/block device: a filesystem may be made on it, mounted from it, or read and write requests can be issued directly to the logical disk. This logical disk device is called a **Logical Volume**.

Logical Partitioning

The storage used by each **Logical Volume** comes from an allocation pool called a **Volume Group**. Each **Volume Group** is a collection of physical disks.

There are no restrictions on the types of disk devices (e.g. SCSI, HP-IB HP-FL) that can be placed in a given volume group, except that HP-IB can not be mixed with non-HP-IB devices in one volume group. SCSI and HP-FL devices can be mixed in a volume group.

Since **Logical Volumes** can get space from any physical disk in the **Volume Group**, a **Logical Volume** may be larger than any one of the underlying disk drives. Hence the size of file systems or databases residing on **Logical Volumes** are not restricted by the size of the physical partitions.

The LVM also provides the capability to expand each **Logical Volume**. This permits flexible allocation of disk resources.

Mirroring

Data stored on a **Logical Volume** may be mirrored for high availability, high read performance, or both. The system may be configured with each **Logical Volume** **singly-mirrored** (two copies) or **doubly-mirrored** (three copies). Each copy of data can reside on different physical volumes or even different disk types. Mirroring is not available on HP-IB disks.

By replicating data, the LVM is able to transparently recover from the loss of one copy or two copies of the data. When the LVM is unable to access a copy, the LVM redirects I/O intended for the missing data to a secondary or tertiary copy.

For applications that have a significantly higher proportion of disk I/O for reads than writes, mirroring a Logical Volume may result in higher performance than would be attainable without mirroring.

Mirroring is not bundled with the base LVM. It can be purchased through the LVM-MIRRORING product.

Bad Block Relocation

The LVM manages the replacement of defective disk blocks within the volume group. This feature is configurable.

When the LVM detects a soft (correctable) read error, it can rewrite the data to the physical drive, potentially correcting the error. Recurring soft errors are treated as a hard error.

When a hard (uncorrectable) error occurs, the LVM can relocate the block using a pool of data blocks that it maintains for this purpose. Future I/O is directed to the new block. If mirroring is in use, then the LVM redirects the failed read to another copy of the data, and then rewrites the relocated bad block. This procedure causes the failed block to be completely invisible to the application layer, and restores full mirror replication. If no alternate copy of the data is available, then the LVM returns a read error.

The LVM can also cooperate with smart disk controllers and drivers to perform hardware bad block relocation. Prior to performing software relocation, the physical disk drive attempts to relocate and rewrite the defective block. If this procedure succeeds, the disk drive again appears defect-free, and the LVM removes its copy of the defect record.

HP Value Added Features

OSF's LVM is available to all OSF members. All members that adopt OSF's LVM have core LVM features in their disk management subsystem. HP-UX has further enhanced HP-UX/LVM as described in the following sections.

Root, Primary Swap, and Dump Logical Volumes

The OSF base LVM code does not provide the ability to put the root, primary swap, or dump volumes on logical volumes. HP has added this functionality to provide the following features.

- **Mirror Root and Primary Swap.** This allows users to increase the availability of their systems by ensuring that a single disk failure cannot crash the system.
- **Common Disk Storage Management.** Root and primary swap may be managed as logical volumes just as are other file systems and user data. Users can avoid the potential for confusion when their data disks are set up and managed with LVM, but root and primary swap are not.

The design for implementing root, primary swap and dump on LVM disks is based on the following limitations:

- The root volume can not be extended. Since most growth of system files is not in the directories that must be part of the root file system (`/`, `/bin`, `/etc`), this should not pose a problem. Growth occurs mostly in directories that may reside on other file systems (e.g. `/usr`), which themselves could reside on other logical volumes.
- For many of the same reasons, the root volume should be relatively small (near to or less than 100 MBs).
- The root file system can not span disks but can be mirrored on other disks.
- Other filesystems such as `/usr` and `/tmp` are separate logical volumes in the root volume group.
- Primary swap can not span disks. Secondary swap volumes can be defined, filling the same need. Primary swap can be mirrored.
- Dumping system core on LVM physical volumes is supported.
- The root logical volume cannot use bad block relocation.

Booting from a Logical Volume

To locate system volumes such as root and primary swap on logical volumes, the kernel was made able, at boot time, to activate the volume group to which the logical volumes belong. It must also be able to select which logical volumes to use as root, primary swap, and dump. The information required to do this is stored in the **Boot Data Reserved Area** on the boot disk and is managed by the `lvlnboot` command.

The locations and sizes of these **system volumes** (root, primary swap and dump) are defined on each disk to be used. Again **lvlnboot** manages this information.

The Boot Disk

The LIF volume is used in much the same way as with non-LVM boot disks. ISL and HP-UX are stored there, as well as the autoboot file, IOMAP, RDB, and so on. In addition, the LABEL file used by SwitchOver/UX is now a standard part of the boot LIF area. Switchover stores in the LABEL file the identification of the primary system locking the boot disk. LVM uses this file to store logical volume information for the root, swap or dump logical volumes on that disk. The LIF file LABEL is created and maintained by the **lvlnboot** command. The LIF volume is set up by **install** and by **mkboot(1M)**.

When referencing the boot disk, partitioning is not available; therefore, the use of Section 2 on the root disk is the only way to access the Boot Disk directly. This is true for any LVM Physical Volume.

LIF file LABEL

The LABEL file contains the address and size of each of the root, primary swap and dump logical volumes which reside on the same disk as that LABEL file. It also contains a flag for each entry defining whether that entry refers to root, primary swap, or dump.

The kernel loader (HP-UX) uses the LABEL file to locate the root file system which where it then searches for the HP-UX kernel.

HP-UX was modified to allow the system to be booted even if the root and/or primary swap logical volumes cannot be configured. This is the case during installation, when memory constraints prevent the LVM from being configured into the kernel. The support tape has similar constraints. In order to boot under these constraints, each disk containing any of the system logical volumes must have a LIF file named LABEL which specifies the location and size of the root logical volume on that disk. These locations are used during a **maintenance mode boot**.

Maintenance Mode

Whenever the LVM data structures are not available (during installation or after disk corruption), the system cannot be booted normally. An alternative boot mode is provided that does not rely on LVM. When a **maintenance mode** boot is selected, the kernel uses the locations found in the LIF file LABEL to locate the root file system and primary swap, thereby removing all dependencies on LVM disk-based data. **Maintenance mode** is used by administrators and support engineers. It does not support multiple users as it implies single-user mode. (Running **init 2** causes unpredictable behavior).

After being booted in **maintenance mode**, the system can be used to create/recover the required LVM disk data.

Changes to S800 File

The **lvvol** keyword may be used in the **uxgen** input file to indicate the intent that a logical volume be used for root, primary swap, or dump. This keyword must be used in the dumps statement if a combination of logical volumes and disk sections are to be used for dump.

For example:

```
root on lvvol;
```

```
swap on cio_ca0.hpfl0.disc2 at 4.2.1 section 13;
```

```
dump on default;    #forces dump to primary swap device
```

or

```
root on lvvol;
```

```
swap on lvvol;
```

```
dumps on lvvol;
```

In the first example, root is a logical volume while swap is on Section 13 of 4.2.1. Therefore, the disk at hardware path 4.2.1 is not a disk belonging to a Volume Group. When using LVM, the location of root, swap, or dump is kept in the LIF file LABEL, not in the kernel itself. To display the locations of the root, swap or dump volumes, the **lvlnboot** command can be used.

3-92 Topics Related to the HP-UX System

In the second example, the root, swap, and dump are all Logical Volumes. This is the recommended configuration.

ISL command line override to LVM volumes

The location of root, swap or dump can be overridden in the same manner as 8.0 with the -aR -aS and -aD options. The -aR option can also be used to select an alternate LVM boot disk.

Booting from Support Tape

The support tape can be used to boot the system if the boot volume is damaged. To make this possible, the location of the root logical volume is at a fixed place on the disk. This limitation requires the root logical volume to be the first logical volume created on a disk. This allows the support tape to look at a fixed location on the disk to examine and/or repair the root file system.

Disk Device Driver Changes

To support the LVM, `disc3` (HP9000/S800 Logical Device Manager(LDM) for SCSI disk), `disc30` (HP9000/S800 Direct Access Device Manager(DM) for SCSI disk), and `disc2` (HP9000/S800 CIO HP-FL disk driver) were modified as follows:

- Change the disk powerfail recovery algorithm for the mirrored LVM IO request to 0.
- Support a new error code for the LVM's bad block relocation.

Offline Access to LV

HPUX DataPair (disk mirroring) allows users to temporarily bring a copy of mirrored section offline and do activities such as backup from offline copy. Before the operation system brings a section offline, the system syncs all filesystems, frees that section, and if that section is a mounted and a clean filesystem, the operating system marks that filesystem clean.

LVM provides similar functionality through the `lvsplit`, `lvmerge` commands. These commands are provided through the LVM mirroring product named LVM-MIRRORING.

Mirror Consistency Recovery Configuration

The OSF provided MWC (Mirror Write Cache) is the mechanism provided by LVM to keep track of outstanding mirror write requests. On a system reboot due to crash or power failure, MWC is consulted to recover mirror consistency due to incomplete mirror writes. While MWC does take care of the data block consistency problem, it may also introduce one extra disk write for every mirror write, which degrades system performance significantly.

To provide better mirroring performance, two new mirroring configurations have been added.

- Mirror with MWC disabled. A mirrored LV with MWC disabled does not use MWC. Upon reboot, LVM does not try to keep all mirror copies of LV consistent. For example, since databases use their log files to recover data files, databases may choose to configure database log files as mirrored LVs with MWC enabled and configure database data files as mirrored LV with MWC disabled. This gives data files both mirroring capability and better write performance.
- Mirror with MWC disabled and mirror consistency enabled. This gives a mirrored LV better write performance and mirror consistency recovery upon system crash or power failure. A new mirror consistency recovery method is used for this mirror configuration. This new method takes much longer time to complete because, without a MWC, each extent in the LV should be assumed to be inconsistent and need to be recovered. However, a daemon is spawned to synchronize the LV in the background, thus making the LV available to the user sooner than otherwise.

VG Import Utilities

The design of current LVM commands is dependent on the existence of the `/etc/lvmtab` file. Each configuration update or query requires a lookup to the `lvmtab` file. If users accidentally destroy the `lvmtab` file, any future LVM configuration related operation is not possible. Since each PV's physical device file name is recorded in the `lvmtab` file, which is not editable, users can not move around their disk connection without going through a **backup**, **unconfig**, **reconfig**, and **restore** VG process. Also, the existing LVM design does not address how to access a VG that was previously created on other computer system. Therefore, the commands **vgexport(1m)**, **vgimport(1m)** and **vgscan(1m)** are designed to achieve the following:

- **Disk reconfiguration:** When it comes time for users to change their hardware connections, the PV path names stored in the `/etc/lvmtab` file after reconfiguration may no longer match to their real PV locations. The `lvmtab` file must be rebuilt to reflect the new disk configuration.
- **Recovery from loss of `/etc/lvmtab` file.**
- **Importing/Exporting a VG:** This is important to support removable media and disks connected to multiple systems.

vgcfgbackup and vgcfgrestore

To recover from such things as disk corruption and a disk failure, backup utilities are sufficient to backup user data on a logical volume. However there is also a need to backup the LVM configuration. The **vgcfgbackup** utility is used to backup LVM configuration for volume groups in binary files (one for each volume group). The **vgcfgrestore** utility enables restoration of the configuration to individual physical volumes.

Integration of LVM with `/usr/bin/sam`

In HP-UX 9.0, LVM features have been added into the System Administration tool that let users add Physical Volumes into LVM, create filesystems during LVM, create logical volumes, create volume groups, and migrate existing physical volumes to logical volumes.

Coexistence with Switchover/UX

Switchover/UX was enhanced to recognize a primary system with a Root Logical Volume. The `SwitchInfo` file has also been enhanced to allow for the possibility of 3-way mirroring of the root logical volume.

Implementation of LVM

LVM stores information in data structures at the top of the Physical Volume. These structures contain the LVM information for that Physical Volume, Boot data, Volume Group information, and other information necessary to construct the Logical Volume layout.

The non-Boot disk contains the following information:

- **Physical Volume Reserved Areas (PVRA)**, contains LVM information specific to that Physical Volume. It is created by the `pvcreate(1M)` command.
- **Volume Group Reserved Area (VGRA)**, contains LVM information specific to the entire Volume Group. A carbon copy of the VGRA is found on each Physical Volume in the Volume Group. Within the VGRA is the **Volume Group Status Area (VGSA)**, which contains quorum information for the Volume Group. The VGRA is created by the `vgcreate(1M)` command.
- **User Data Filesystems** and **raw logical volumes** are all contained on Logical Extents. The map of Logical Extents is contained in the VGRA.
- **Bad Block Relocation Area** contains information specific to the bad block recovery mechanism.

The Boot disk contains the following information:

- **LIF header** contains the LIF header information. This area is also present on Physical Volumes containing dump or swap devices. It is created by `mkboot(1M)`.
- **PVRA** see above.
- **LIF Directory Area** containing HP-UXboot, LABEL files and other LIF files. Created by `mkboot(1M)`. LABEL files created by `lvlnboot(1M)`.
- **VGRA** see above.
- **User Data Root Filesystem** must be the first Logical Volume.
- **Bad Block Relocation Area** see above.

The following header files have been added to `/usr/include`:

<code>lvm/l_defect.h</code>	Describes bad block directory structure
<code>lvm/lvm.h</code>	Defines all the LVM <code>ioctl()</code> commands and data structures.
<code>lvm/pvres.h</code>	Describes the PVRA and LVM record
<code>lvm/vgres.h</code>	Describes the VGRA
<code>lvm/vgsa.h</code>	Describes the VGSA
<code>lvm/bdres.h</code>	Describes the BDRA.

Changes have been added for LVM in the following header files:

`sys/dsk_label.h` (Describes the LIF file LABEL)

These header files are provided for reference only. Modifications to the LVM data structures on the Physical Volume or direct `ioctl()` calls to the LVM pseudo-driver are not supported.

Terminology

The LVM introduces new terms to the HP-UX system. The terms are described in the glossary for the *How HP-UX Works: Concepts for the System Administrator* document.

LVM Command Interface

LVM provides numerous commands to configure and manipulate Logical Volumes. These new commands are described in the HP-UX Reference.

<code>extendfs</code>	Extend an offline filesystem.
<code>lvchange</code>	Changes the characteristics of a logical volume.
<code>lvcreate</code>	Creates a logical volume in a volume group.
<code>lvdisplay</code>	Displays information about logical volumes.
<code>lvextend</code>	Increases the number of physical extents allocated to a logical volume.
<code>lvlnboot</code>	Prepares a logical volume to be a root, swap or dump volume.
<code>lvmerge</code>	Merges previously mirrored volumes into one logical mirrored volume.
<code>lvreduce</code>	Decreases the number of physical extents allocated to a logical volume.
<code>lvremove</code>	Removes one or more logical volumes from a volume group.
<code>lvrmboot</code>	Removes a logical volume link to root, swap or dump volume.
<code>lvsplit</code>	Splits a mirrored logical volume into two logical volumes.
<code>lvsync</code>	Synchronizes logical volume mirrors that are stale in one or more logical volume.

pvchange	Changes the characteristics of a physical volume in a volume group.
pvcreate	Creates a physical volume that can be used as part of a volume group.
pvdiskdisplay	Displays information about one or more physical volumes within a volume group.
pvmove	Moves allocated physical extends from one physical volume to another.
vgcfgbackup	Saves LVM configuration for volume group.
vgcfgrestore	Restores LVM configuration onto the volume group.
vgchange	Sets the status of a volume group to on or off.
vgcreate	Creates a volume group.
vgdisplay	Displays information about volume groups.
vgextend	Extends a volume group by adding physical volumes to it.
vgexport	Exports a volume group from a system.
vgimport	Import a volume group onto the system.
vgscan	Scans the system's physical volumes for volume groups.
vgreduce	Reduces a volume group by removing one or more physical volumes from it.
vgremove	Removes the definition of one or more volume groups from the system.
vgsync	Synchronizes logical volume mirrors that are stale in one or more volume groups.

Impact

- Since LVM allows a large amount of flexibility in the size of filesystems with regard to the Physical Volumes, up front planning should take place before configuring for LVM. For example, if a filesystem is to be mirrored, the user may wish to place the mirrored copies on separate disk controllers.

3-98 Topics Related to the HP-UX System

- **Entire Physical Volume is Utilized:** The sectioning scheme for Physical Volumes is no longer used under LVM. The entire disk (section 2) must be used for LVM Physical Volumes.
- If the user wishes to use LVM for the Root filesystem, systems migrating from a prior release of HP-UX must take a full backup of each filesystem including root which is converted to a Logical Volume. The HP-UX install application (not update) is then run to configure the Root Logical Volume. The system then reboots onto the Root Logical Volume and allow the update to continue. Once the installation is complete, the LVM commands or `/usr/bin/sam` may be used to convert the other previously backed up filesystems to Logical Volumes.
- LVM replaces the mirroring capability of the DataPair product. LVM supports 2- or 3-way mirroring and mirroring on HP-FL or SCSI disks. HP-IB is not support for mirroring. All efforts should be made to migrate existing DataPair filesystems to LVM. This can be done through the LVM mirroring product named LVM-MIRRORING.
- If an application wishes to access Logical Volumes where it previously accessed the Physical Volume, the application must use a new device pathname.

DataPair Features

To provide the same functionality as offered by DataPair, the following features were added to the LVM and are provided in the LVM-MIRRORING product.

- **Mirrored root and primary swap volumes.** To allow high availability systems to survive single disk failures, the ability to mirror the root and primary swap volumes is required.
- **Disk Powerfail.** If a disk becomes unavailable due to a power failure or unexpected disconnection, mirrored logical volumes should switch to another copy of the data.
- **Offline access to a copy of a logical volume.** This provides the ability to take a snapshot of a logical volume for later backup. If a file system resides on such a logical volume, consistency of the backup copy of the file system is maintained.
- **Read-Only VG**

- **Mirrored write performance.** DataPair never requires additional (log) writes when writing to a mirrored logical volume. A logical volume configured with MWC disabled and mirror consistency enabled provides a similar function.

Performance

When configured appropriately for a system, the additional overhead of LVM on a filesystem or raw device is minimal.

Size Requirement

LVM commands occupy 2 MBs of space in */etc*.

A default volume group uses about 400 KBs of disk space at the top of each Physical Volume, and rounds the remaining space down to the nearest MB.

Memory

Kernel Support for Large Memory

Summary of Change

The 9.0 kernel can address up to 2 GBs of main memory.

Impact

Very large time sharing and server systems may now be configured.

Computers Affected

The only computers that support 2 GBs are the Series 800, Model 890 and Model x00.

Buffer Cache

Changes to bufpages Between 8.0 and 9.0

The buffer cache interprets the value of **bufpages** as the number of physical pages to use for the buffer cache. At 8.0, all machines had a page size of 2 KBs or 2048 bytes. A value of 1024 for **bufpages** yielded buffer cache of 2 MBs. At 9.0, the size of a page has increased to 4 KBs or 4096 bytes, which doubles the size. The value of 1024 used in 8.0 now generates a buffer cache of 4 MBs.

Besides increasing the space by two, the number of buffer headers increased. Because of the increase in page size, the number of headers is now set equal to the number of pages. The default is to assign 4 KBs of space to each header in both 8.0 and 9.0. The difference is that 8.0 has a 2 KB page size and 9.0 has a 4 KB page size.

8.0 genfiles that specify **bufpages** allocate twice as much space in 9.0. To use the same amount of buffer space as in 8.0, cut the value of **bufpages** in half.

Larger Buffer Cache Configurations

At 8.0, the largest buffer cache configuration was about 53 MBs. At 9.0, the buffer cache was changed to support up to 0.9 GBs of physical space. Systems that mix file system block sizes and/or large buffer cache configurations should consider potential impacts.

Summary of Change

Prior to 9.0, the size of the buffer cache was restricted by the kernel virtual space, physical memory in the system, and file system block sizes. When the physical size of the system reached approximately 450 MBs, the default size of the buffer cache began to shrink. At 8.0 the largest possible buffer cache was about 53 MBs. The 9.0 release removed some of these restrictions to allow larger buffer cache configurations.

The buffer cache no longer preallocates all the virtual space. In the default case, 10% of physical memory is used. The buffer cache creates one buffer header for every physical page in the buffer cache. Each buffer header is then given one virtual and one physical page. The default configuration satisfies all requests one page in size or smaller (at 9.0, a page is 4 KBs).

When a buffer is used for a request greater than its virtual or physical size, it must be grown. The physical space is acquired by stealing pages from other buffer headers. The virtual space is acquired from the buffer cache's virtual space pool, while the space originally held is released to be used for another request. Buffer headers retain the increase in virtual space unless the total buffer cache virtual consumption reaches the buffer cache high water mark(.9 GBs).

The buffer cache has a virtual size restriction and the high water mark is an indicator that the total consumption is too high. When this occurs, the buffer cache no longer allows buffers to retain the excess virtual space they may have. Buffer headers are allowed to grow or migrate their virtual space to the largest file system block size. (Physical space is just allocated back and forth amongst the headers.) This way any buffer header can be used for the largest size request. This works until the (virtual) buffer cache is so big that the number of buffer headers multiplied by the largest mounted file system block size is greater than the high water mark. Most buffer caches should not exceed the restriction, but there are systems requiring larger buffer caches, hence larger virtual demands.

After the high water mark is reached, buffer headers are not allowed to keep excess virtual space. Instead, as the buffer header is allocated, any excess space is returned to the buffer cache virtual pool. The trimming back of buffers continues until the buffer cache virtual consumption becomes less than the low water mark (.8 GBs). This shrinking of the buffer headers is extra system overhead that causes some system degradation. The amount of degradation depends on the size of the buffer cache and the mix of the file system block sizes.

Impact

Because each buffer header does not have all the virtual space preallocated, certain configurations may be more prone to performance problems. Smaller buffer cache configurations should not have any degradation problems.

For instance, a 500 MB system using the defaults (not configuring **nbuf** or **bufpages**) would have 50 MBs of buffer cache. If that system had only 8 KB file systems connected, the buffer cache would have enough virtual space to accommodate all the buffer headers. The buffer cache would never need to shrink any buffers. It only needs to manipulate the physical space. No degradation would occur.

3-102 Topics Related to the HP-UX System

If a system has 1.5 GBs of memory and a 800 MB buffer cache configured and if the system has only 4 KB file systems, then the buffer cache would never exceed the high water mark. However, if one or more file systems of block size 8 KB are mounted, the buffer cache would try to grow every header to 8 KB of virtual space. The total amount of virtual space would exceed the maximum. The system would then force buffer headers to release their excess virtual space. If this happens often, the buffer cache is not running at optimum performance and may be slower than a buffer cache of smaller size.

A better choice for the latter configuration is to also configure the number of buffer headers (**nbuf**). By default, for every page of physical space, one buffer header is created. In the case of the 800 MB buffer cache, there would have been as many buffer headers as pages. Then, 8 KB file systems should need fewer headers as they buffer more space, assuming they are not accessed frequently for fragment sizes. If **nbuf** is reduced, the buffer cache could have grown more buffer headers before exceeding the high water mark. The reduction in the amount of headers would have decreased the amount system overhead.

Once the high water mark was exceeded, buffer header virtual reduction continues until the buffer cache virtual consumption is less than the low water mark of 0.8 GBs. Depending on the configuration, some systems may always stay in the reduction mode. For instance, a system configured with a 900 MB buffer cache and 8 KB file systems could get into this state. As 8 KB requests come in, the buffer cache would exceed the high water mark and the buffer cache would then begin to reduce virtual consumption. Since each header was only given one page, the minimum request size is one page. The 900 MBs of physical space is close to the high water mark and forces a one to one correspondence for virtual and physical space.

Large buffer caches should choose **nbuf** and **bufpages** appropriately.

Performance

As discussed in the impact section, certain buffer cache configurations may be prone to degradation. Users of smaller buffer caches with 8 KB file systems should not see any problem. Only users who mix file system block sizes with large buffer caches or large buffer cache configurations (greater than 600 MBs) have a potential for degradation.

A measurement to see if your system exceeds the high water mark is to multiply the number of buffer headers in the system (**nbuf**) times the largest file system block size. If the **nbuf** parameter is not configured via a tunable parameter, **adb** can be used to extract its value once the system has booted. The **nbuf** parameter is static and does not change. If the **nbuf** parameter multiplied by the largest file system block size is greater than the high water mark (0.9 GBs), then your system has the potential for overhead due to exceeding the high water mark. There is no guarantee this occurs. It depends on each system configuration. The virtual consumption can be reduced by configuring **nbuf** and **bufpages**.

Availability of Lockable Memory on an 8-MB System

On a system with 8 MBs of memory, if the default values of configurable parameters as specified in **/etc/master** and **/etc/conf/gen/S800** are used to generate the kernel, the system may not have enough memory available to be used as lockable memory.

If a user needs to use lockable memory, it is recommended that the **maxusers** parameter be reduced to a value no greater than 16. The size of many kernel data structures are determined as a function of **maxusers**. Reducing the value of **maxusers** allows memory to be freed and used as lockable memory.

To reduce the value of **maxusers**, add the following line in the **/etc/conf/gen/S800** file and regen a kernel:

```
maxusers XX;
```

Substitute XX with the new value chosen for **maxusers** (for example, 16).

Equivalently Mapped Memory

The eqmemsize Tunable Parameter

There is a new tunable parameter, **eqmemsize**. This tunable parameter defines the size, in pages, of the equivalently mapped memory reserve pool. This tunable parameter should only be changed in the event the system is failing due to a lack of equivalently mapped pages.

Knowing when the system is out of equivalently mapped memory is not easy. The system tries to dynamically handle equivalently mapped memory, so in general users should not have to worry about modifying this parameter.

Summary of Change

Equivalently mapped memory is used by drivers and subsystems which need a page whose address is the same in both real and virtual mode. Drivers use these pages to transfer information between hardware interfaces. In virtual mode, the driver places data on the page. In real mode the hardware transfers the data using DMA (Direct Memory Access), bypassing the translation. Because the virtual and real address are the same, no special processing need occur.

During the boot process, the I/O system builds a pool of equivalently mapped pages based upon the I/O configuration.

If for some reason the anticipated need is not met, the I/O subsystem has other options from which to obtain equivalently mapped pages. One of the options is the equivalently mapped memory pool. The equivalently mapped pool has a defined size of **eqmemsize** entries. Currently the default is 10 pages. One of these pages can be used, but it first attempts to convert an existing free page into an equivalently mapped one. Depending on the size of the machine and the load level, pages may be more easily converted than others. So the equivalently mapped pool has a minimum of **eqmemsize** pages, but with the conversion, there is the possibility of a whole lot more.

Because the equivalently mapped memory allocator tries to convert pages, it should rarely be the case that the system runs out of equivalently mapped memory.

Impact

Systems with very large I/O configurations or large amounts of X.25 cards may need to increase **eqmemsize**. If the system experiences a panic due to the lack of equivalently mapped memory, **eqmemsize** should be increased. Having this failure should be rare as the reserve pool is used only if a memory page cannot be converted. Larger systems, those exceeding 1 GB of physical memory, have a better chance of failure.

The pages placed into this pool cannot be used for any other purpose other than equivalently mapped requests. For this reason, the value of `eqmemsize` should not be placed arbitrarily large, since these pages cannot be used for normal system processes.

Configuration

The tunable parameter `eqmemsize` indicates how many pages should be placed into the equivalently mapped allocator. (The default is 10.) The value should only be increased if a system failure due to the lack of equivalently mapped memory exists. If the value needs to be increased, then do so based upon the I/O configuration of the system. Larger configurations require greater increases than smaller systems. For X.25 users, there should be one page for every X.25 card in the system.

There is another tunable effecting equivalently mapped memory, `iomemsize`. This configures the I/O systems static emergency pool. This value should not be changed for problems dealing with lack of equivalently mapped memory. While it applies to the majority of equivalently mapped users, it does not cover all users.

Pseudo Swap

Features

This allows the system to use memory for swap space after physical swap space was exhausted. With very large memory systems, it becomes less desirable to have enormous amounts of disk space for swap. Using pseudo swap, this requirement is relaxed.

Summary of Change

It was a requirement for a process using N pages of memory to reserve N pages of swap space. This requirement made the amount of swap space the limiting factor in determining the total number of processes that could be active at any one time. For example, a system with 64 MBs of memory and 32 MBs of swap space could run only processes that totalled less than or equal to 32 MBs. Additional processes could not start because they could no longer reserve swap.

Pseudo swap was designed to use system memory for swap after the physical swap space was exhausted.

The amount of memory available as pseudo swap is equal to 3/4 of available memory after boot. Available memory is the memory that remains after the kernel has booted. This amount is reported in the message buffer during boot. Pseudo swap is limited by the amount of total lockable memory. Total lockable memory is the amount of memory users can allocate using `plock()` calls. The value of total lockable memory is determined dynamically or when the user specifies a value for `unlockable_mem` (See System Administrators manual). Like available memory, the amount of total lockable memory is reported to the kernel message buffer and to the system console during the boot process.

Impact

Applications referencing region structures defined in `sys/region.h` need to be recompiled.

Configuration

The tunable `swapmem_on` parameter is set to 1 by default, indicating `swapmem` (pseudo swap) is enabled. Setting this tunable parameter to 0 and rebuilding the kernel disables the pseudo swap feature.

Size Requirement

The amount of memory used for pseudo swap is calculated dynamically during boot and is based on the amount of free memory. It is not user configurable. The amount of memory used for pseudo is equivalent to 3/4 of the available memory to the users. Available memory is reported by the kernel during the boot sequence.

Networking

The netmemmax Tunable Parameter

There is a new tunable parameter, **netmemmax**. This tunable parameter defines the size of the dynamically malloc'able memory for IP level reassembly of packet fragments. This tunable parameter, like other tunable parameters, can be modified in the gen file. This tunable parameter can be changed if the system response degrades to a point where it appears to hang.

Summary of Change

After stressful networking activity, mostly involving UDP applications, a system's response might degrade to a point where it appears to hang. This can happen when memory was fragmented into small chunks that are not handled by garbage collection and compaction processes that make the chunks available to user-space applications. Under these conditions, a user can tune HP-UX with **netmemmax**.

Impact

At 9.0, the default value for **netmemmax** is 10% of the dynamic memory. This value should be sufficient for most user applications that work under normal conditions and network configurations. Previously there was no limitation on memory usage.

Configuration

When netmemmax == 0	The kernel may use at most 10% of dynamically malloc'able memory for IP level reassembly of packet fragmentation. (9.0 release default value)
When netmemmax == -1	The kernel places no restriction on dynamically malloc'able memory for IP level reassembly of packet fragmentation. (8.0 release default value)
When netmemmax == X	The kernel may use at most X bytes of malloc'able memory for IP level reassembly of packet fragmentation.

Features

The default buffer size for the TCP stream socket has changed.

Summary of Change

The maximum buffer size for a TCP stream socket is 58254 bytes. The default buffer size was increased from 4096 bytes to 8192 bytes. The send and receive buffer sizes for TCP stream sockets can be altered by using the `SO_SNDBUF` and `SO_RCVBUF` options of the `setsockopt()` system call.

Impact

Customer applications which call `setsockopt()`, after the connection is established, to set the send or receive buffer sizes to less than 8192 bytes, fails. This is to avoid shrinking of TCP windows. Customers should set the send/receive buffer sizes (if decided to reduce the default value), before the connection is established.

Performance

The increase of the send and receive buffer sizes (default values) improves performance for fast links.

Partitions (Includes Filesets)

A release of HP-UX is distributed on media such as a CD-ROM or DDS cartridge tape. In some cases, you might be able to obtain media via a networked server. In either case, the media contains Partitions, which are collections of filesets. Filesets are collections of files that relate to a major functionality. You can get a list of available Partitions, and their Filesets, in the *Installing and Updating HP-UX* document.

New SLIP-RUN fileset

Summary of Change

The NETINET fileset does not include the Serial Line IP (SLIP) feature. Instead, there is a new fileset, SLIP-RUN, which provides this feature.

Impact

To use the SLIP functionality, you should update your system by installing the SLIP-RUN fileset. You also need the NETINET fileset.

XNS Fileset

The XNS fileset was removed from the 9.0 Release.

NETTRACELOG Fileset

Features

The NETTRACELOG fileset, consisting mainly of the **nettl** and **netfmt** commands, is essentially the same as for earlier releases. All options and behaviors have been preserved.

Summary of Change

The most significant change to the HP-UX 9.0 version of the formatter program, **netfmt**, is the use of shared libraries. Subsystems now provide subformatters to decode their tracing and logging information in the form of a shared library. This allows the **nettl** and **netfmt** commands to dynamically include subsystems from any products installed on the system. There is no need to relink **netfmt** and **nettl** each time a new product is installed or removed. The commands are no longer recreated on the system. Therefore, the **nettlgen** command was obsoleted.

The NETTRACELOG configuration file, **/etc/conf/nettlgen.conf**, has been revised. There is a new field, subsystem type, which determines whether the subsystem is from the kernel or user space. This information is necessary to support the various kernel space products that use tracing and logging.

The fields for the NS subsystems and the master formatter have been changed to use defaults. These changes are transparent to the user since the NETTRACELOG customize script converts the `nettlgen.conf` file from the old format to the new.

Another change to the `nettlgen.conf` file is the addition of the console option file that controls console logging. This gives a system administrator more control over the logging information that gets sent to the console, if any. The console option file behaves just like other filter files. A default file is created whenever no console option file is specified.

Although customers should never have to look into the configuration file, it is possible that they might. The file is self documenting, with each field's use and permissible values described at the beginning.

These new changes are further described in the entries in the HP-UX Reference for the product.

Impact

The user should not see any difference in the behavior of `nettl` or `netfmt` with the use of shared libraries. Only product developers need to be concerned with the shared library requirement of `nettl` and `netfmt`. Converting from archived to shared libraries is a matter of changing compile and linking options. No code changes should be necessary.

During the migration to this version of `netfmt`, it is possible that not all subsystems are updated at the same time. For example some filesets which are not bundled with the HP-UX 9.0 release may be updated at some time after the update to HP-UX 9.0 has been performed. This leads to a situation where the NETTRACELOG configuration file, `/etc/conf/nettlgen.conf`, contains outdated information. In particular, references to archived libraries are left in the file.

The new version of the `netfmt` command are not able to use the NETTRACELOG configuration file when it contains references to archived libraries. The `netfmt` command can only link with shared libraries. An attempt to load an archived library results in an error. The error may appear on the system console during system startup or whenever `nettl` is started.

When the NETTRACELOG configuration file contains references to archived libraries the offending subsystem configuration line must be removed. This can

be done using a text editor on the file `/etc/conf/nettlgen.conf`. The entire line may be deleted or commented out using the `#` character. It is assumed that the outdated subsystem automatically reconfigures itself when the proper version has been installed on the system.

In general this situation can only occur when filesets that are not bundled with the current release of the operating system have not been updated.

The syntax change to `nettlgen.conf` should have no impact to the user since the `nettlconf` and `customize` scripts should do any conversion necessary.

Performance

Using shared libraries may be slightly slower than the old method since the libraries have to be dynamically loaded each time a command is used. However, since `nettlgen` was obsoleted there is time saved in forgetting to re-run `nettlgen` and making other similar mistakes. The `nettlgen` command itself was extremely slow.

Alternatives and Compatibility

All trace/log files taken on older systems (8.0 and earlier) should be forward compatible and readable by `netfmt`. All trace/log files taken on 9.0 systems should be formatable on older versions of `netfmt` provided the individual product subsystem subformatters supports this.

Obsolescence

The `/etc/conf/nettlgen` command was replaced by the use of shared libraries. The `nettlgen` command took a long time to execute and was prone to errors.

Subsystem formatters in the form of archived libraries are no longer usable by the `netfmt` command. All subsystem formatters must be provided in the form of shared libraries so that they may be dynamically loaded by the `netfmt` command as needed.

XNS Fileset

The XNS fileset was removed from the 9.0 Release.

SCSI Information

This section contains information (hardware and software) that relates to SCSI.

Additions to the SCSI ioctl Interface

Two new SCSI specific `ioctls` have been added to the SCSI tape driver. They are `SIOC_INQUIRY` and `SIOC_XSENSE`.

The `tape14` driver does not provide a pass-through interface as the `s2tape` does. This, in combination with the `tape14`'s state computer, prevents the application from getting current REQUEST SENSE data in all situations.

The `error_code` field in the REQUEST SENSE structure reflects the freshness of the data. A `0x70` in the `error_code` means that the request sense buffer results from an asynchronous request without a prior check condition. Error code `0x71` indicates request sense data saved from the last non-benign check condition.

Shared Libraries

New Features

The following routines and options were added in the 9.0 release. For more information, see the `shl_load(3X)` entry in the HP-UX Reference and the 9.0 Programming on HP-UX document.

New Routines

`shl_getsymbols()` Used to return the list of symbols from a shared library

New Options

SHLIB_PATH	Environment variable that can be used to specify directories to search at run-time to locate shared libraries needed by a program.
shl_load()	DYNAMIC_PATH flag to indicate the library path passed in to the routine is subject to dynamic directory search specified by SHLIB_PATH environment variable.
shl_load()	BIND_RESTRICTED flag to indicate that the symbols available to the library being loaded should be restricted to those present at load-time.

Additional New Routines

Additionally, users who move directly from HP-UX 8.0 or 8.02 to HP-UX 9.0 see the following new features.

shl_definesym()	Used to add a symbol definition to the shared library system.
shl_gethandle()	Get information about the given handle

Additional New Options

Additionally, users who move directly from HP-UX 8.0 or 8.02 to HP-UX 9.0 see the following new options.

shl_load()	New options BIND_FIRST, BIND_NONFATAL, BIND_NOSTART, BIND_VERBOSE
------------	---

Shared Library Global Data

When a program file accesses a data item defined in a shared library, the linker copies the data item to the program file at link-time. Since the program file contains a static copy of the initialized data, changes to the initialization value for the shared library data item were not seen prior to 9.0 unless the program was relinked. At 9.0, the linker generates records that allow the shared library loader to update initialized data items found in the program file at run-time. The loader uses the initialization value found in the run-time shared library.

Impact

Programs linked with the 9.0 version of `/bin/ld` and run on a system with the 9.0 version of the shared library loader `/lib/dld.sl` experience run-time initialization of data items that are defined in shared libraries.

Signal Change

In 8.0, when `dld.sl` (the shared library loader) terminates a process, it sends the signal `SIGTERM`. The entry for `dld.sl(5)` has always stated that the loader sends the signal `SIGABRT`, and that is what the Series 300/400 loader does. At 9.0, the Series 700/800 loader sends the signal `SIGABRT` when it must terminate a process.

Impact

Once a program that uses shared libraries has started, the only time the termination signal is sent is if a run-time unresolved symbol is encountered (only possible when using deferred binding). Programs that encounter this error and attempt to trap on `SIGTERM` in an attempt to catch it must now trap on `SIGABRT`.

Errno Change

Change applies to users who move from 8.0 or 8.02 to 9.0.

`shl_findsym()` Change in `errno` setting if failure. In HP-UX 8.0, `errno` was set to `ENOSYM` if the symbol could not be found. In 8.05, the following status information is returned on failure:

	return_val	errno	Description
1)	0	0	Symbol was found.
2)	-1	0	Symbol was not found.
3)	-1	ENOSYM	Symbol was found, but a symbol it depends on was not found.
4)	-1	???	Other error.

Impact

Programs relying on the value being placed in `errno` when the routine fails need to track the new behavior.

Smart-Bind

This functionality can impact the migration of certain shared libraries from 8.0 systems to 9.0 if they are not relinked first. As of 8.05, the shared library loader implemented a new binding algorithm called Smart-Bind. This algorithm relies on special tables prepared by the linker. The 8.0 linker actually built these tables in anticipation of this new algorithm, but it had a limitation that surfaced if a shared library was built from assembly language files. Using the assembler, it is possible to directly call one routine from another routine rather than making a PC-Relative call. In this case, the linker did not detect the call and did not properly build the support tables needed by the shared library loader to implement the Smart-Bind algorithm. The symptom is invalid data access at run-time because the data addresses used by the routine being called may not be fixed up. The invalid data access may result in a core dump, but the symptoms vary. This is only a problem on the HP-UX 8.0 and 8.02 releases, and the only library reporting this problem was `libc.sl`.

Workaround

Relink the shared library using the HP-UX 9.0 linker.

C Math Functions and Shared Libraries

For performance reasons, the PA-RISC 1.1 versions of the `libm` and `libM` math libraries are provided as archive libraries only. This can affect an application if C math functions are used in a shared library that is `shl_loaded` by another shared library (the parent library) when the math functions are not used in the parent library. In this case, an application linked on HP-UX 9.0 with the parent library fails at runtime due to unresolved external references to the math functions if and when the other library is `shl_loaded`.

For more information about HP-UX math libraries at 9.0, see the Math Libraries topic in this file and the HP-UX Floating-Point Guide (P/N B2355-90024).

Standards Information

For the 9.0 release, HP-UX conforms to the following standards:

- OSF AES OS Component, Revision A (s300/s400/s700 only)
- X/Open Portability Guide Issue III, Base Branding, including the optional internationalization features.
- FIPS 151-1 and FIPS 151-2
- POSIX.1:1990 (IEEE Standard 1003.1:1990) (also known as ISO/IEC 9945-1:1990)
- POSIX.2:1992 (IEEE Standard 1003.2:1992) (also known as ISO/IEC CD 9945-2.2)
- All options except the User Portability Utilities Option
- ANSI C (ANS X3.159-1989) (also known as ISO/IEC 9899: 1990)
- System V Interface Definition (SVID2)
 - Base System
 - All Extensions except: AS, NS (all platforms), and shmat(KE_OS) (s700/s800 only)
- OSF/Motif 1.2.1
- X Window System Version 11, Release 4
- X Window System Version 11, Release 5

Symbolic Debugger

With the 9.0 release, many languages are available as separate products. The symbolic debugger is included with some of these language products (for example, Pascal, FORTRAN, and C). These products have their own release notes including separate release notes for the symbolic debugger.

The release notes for the products are placed on the system during the update process. They appear, after the update, in the following directory:

`/etc/newconfig/90RelNotes`

System Administration Manager (SAM)

Features

The following items indicate the new or changed SAM functionality:

- OSF/Motif Graphical User Interface.
- Alpha-Terminal interface matching the graphical version.
- Object-action user interaction paradigm.
- User customizable object lists with user selectable sorting and filtering.
- Logging of SAM actions.
- New hardware supported:
 - Models 890, 887 and 897
 - Magneto-Optical disks
 - NIO Fiberlink interface (series 800 only)
 - Combination serial/LAN cards
 - Network printers
 - Compressed DDS and QIC tape devices
 - Fiberlink disk arrays (series 800 only)

■ Modified SAM system areas:

□ Peripheral Devices

Improved configuration of serial and parallel printers. TCP/IP and NPX network printers. Cancel print jobs. Save and restore lp configuration. Remove terminal and modem configurations.

□ Disks and File Systems

Replaces file system management and disk configuration. Logical volume management (series 800 only). Magneto-optical autochangers.

□ Network Configuration

Enhanced and reorganized for improved usability. Networking services status screen. Support for some NFS 4.1 features. X.25 configuration appears only when X.25 is installed.

□ Backup and Recovery

Remote backup device support now more visible. All backup devices on the system are now listed.

□ Kernel Configuration

Access to modification of all kernel parameters. Non-HP drivers are not removed by SAM during re-generation.

■ New SAM system areas:

□ Remote System Administration

Easy execution of SAM on another system.

□ Routine Tasks

Log file trimming. Large and core file removal. System shutdown.

□ Process Management

Process kill and renice. Performance monitoring. Job scheduling via cron.

■ Removed SAM system area:

- Cluster configuration (not supported in 9.00 for the series 800). However, it remains available for series 300/400/700 systems.

Summary of Change

SAM now has a Motif Graphical User Interface (GUI). This change is also apparent in the character-based terminal user interface, which is nearly identical to the new GUI. Upon first entering SAM, users of previous versions are encouraged to explore the Help screens to understand the new user interface and character-based terminal keyboard traversal.

The user interaction model for SAM has changed. In past versions, the user traversed SAM menus knowing what action was desired. At the end of the traversal the user selected the object (or device) to perform the action on. Often the selection of the object required the user to use the help system to present the list of objects (such as disk drives) which SAM could find. With 9.0 SAM, once the user selects an area of interest, such as user management, they are presented with the list of valid objects; i.e. the list of current users. In this way the user sees the current system state up front. The user can sort or filter the list of objects as desired and perform actions on the objects. On selecting an action to perform, users are led to task oriented dialogs that guide them through the task.

The following items note the functional improvements and additions to SAM:

- SAM now logs its actions to a file. The level of this logging is user selectable and the maximum size of the log file is user customizable.
- SAM now performs more rigorous tests when a printer is added.
- Network printers using the TCP or IP and NPX interfaces can now be configured into the spooling system using SAM.
- Print jobs in the LP Spooler can be canceled.
- When a user successfully adds or removes a printer the new state of the printer spooler system is automatically saved. Later, if problems occur because the printer spooler state becomes corrupt, the user can restore the spooler to the last saved state. Users are also able to explicitly save the spooler's state at any time. The state is also automatically saved when updating an 8.0 system to 8.3 or 9.0.
- Terminals or modems currently configured can now be removed.
- The File System management area of SAM was replaced with a new area called Disks and File Systems. This new area incorporates Logical Volume

Management (on Series 800 only) as well as disk configuration and File System Management.

- Magneto-optical disk devices, including optical disk autochangers can now be configured using the Disks and File Systems area of SAM. Also, these optical devices are valid backup devices in the Backup and Recovery area.
- The networking areas (NNC and X.25) have been reorganized. The new sub-areas are: Status, Network Interface Configuration, Remote System Connectivity, Device Connectivity, Network File System (NFS), and Security. The Status sub-area provides information regarding services installed in the system (i.e. Anonymous FTP, Sendmail, NFS, Bootp, NS-NFT, PAD Services and Network Computing System). The Local Location Broker of the NCS subsystem and the supported features of NFS 4.1 can now be enabled/disabled in this sub-area. The Network Interface Configuration section sub-area provides access to IEEE 802.3, Ethernet, Token Ring, FDDI, and X.25 network link information. Remote System Connectivity provides access to Internet Connectivity, Emulation (X.25), UUCP (X.25), and X.3 profile (X.25) configuration screens. PAD Server, PAD Printers, and X.3 Profiles can be configured in the Device Connectivity sub-area. The Network File System sub-area allows configuration of remote and local file systems. Individual directories, instead of whole file systems, can now be exported. The NFS anonymous userid and permissions for file access can now be set.

Note: The portions of SAM regarding X.25 are not present until the X.25 software product is installed on the users system.

- The Kernel Configuration area of SAM does now allow the user to see all kernel parameters, loaded drivers and sub-systems. The users can modify all supported parameters (including the use of expressions) that are validated by SAM. Parameters unknown to SAM can be modified at the users own risk. Any unknown drivers found by SAM are put back in the kernel when it is re-generated.
- The new Remote System Administration area of SAM allows the users to specify remote systems they want to administer. SAM then sets up the connection to these systems so that the user may easily select them at a later date.
- The new Routine Tasks area of SAM contains significant functionality to help the user with the ongoing maintenance of the system. Log file trimming

allows the user to easily identify and trim ASCII and binary system log files. The user may also add application log files to the list. The large and core file removal sub-areas allow the user to easily find and remove unwanted core files, or files greater than a user specified size. Finally, system shutdown provides a simple interface to changing the user state or shutting down the system.

- The new process management area of SAM allows the user to easily view running processes and then renice or kill them if desired. Users can also schedule jobs via cron. Finally, users can start one of the following performance monitors: `/usr/contrib/bin/monitor`, `/usr/bin/top`, and `/usr/bin/sar` (Series 800 only and only if it already exists on the system).
- Users now select a backup device from a list of valid backup devices on the system. The user is no longer required to specify a device file for a local backup device. SAM also supports backing up to and recovering from a remote backup device.

Cluster Configuration area of SAM is not present in the 9.0 for Series 800 computers because this functionality is not supported. However, it remains available for Series 300/400/700 systems.

Impact

All users of SAM are impacted by these changes to SAM. There are two areas of potential user customization of SAM in previous releases: Task Customization and menu additions in Other Utilities.

Customized tasks that the user has placed in the User or Cluster configuration areas of SAM must be re-added when SAM is updated to 9.0.

Menu modifications the user has made in the other Utilities area for 8.0X SAM versions are not compatible with 9.0 SAM. When an 8.0X system is updated to 9.0, these menu customizations are automatically updated to the 9.0 format for the user.

Performance

The interactive performance of 9.0 SAM is generally similar to 8.0X versions. Entry into some of the SAM areas may take longer in the 9.0 version because more information is being presented to the user, and it can take significant amounts of time to collect this information. When significant amounts of time are required for any SAM operation, progress messages are presented to the user.

Alternatives and Compatibility

If a user is dissatisfied with the new SAM, all the administration capabilities of SAM can be accomplished manually using techniques described in the HP-UX documentation.

Some Limitations in SAM

The following limitations might affect your use of SAM.

- The SAM kernel configuration area does not support reading in a new set of initial values from an existing kernel or configuration file (**S800** or **dfile**). This functionality may be documented in the System Administration Tasks manual, but SAM does not support it.
- The SAM kernel configuration area does not support adding, removing, or modifying dump devices on Series 800 computers. Also, SAM does not support modifying swap devices on any Series. Both of these tasks are documented in the System Administration Tasks manual, but SAM does not support them. SAM still supports modifying the console on Series 800 computers.
- The SAM kernel configuration area does not support modifying **cdfs** or **nfs** in a clustered environment as documented in the Managing Clusters manual. Instead of having SAM automatically generate all cnode kernels to match **cdfs** and **nfs** changes made to the server kernel, SAM prompts the user to make the same change on each of the cnodes prior to rebooting the server.
- SAM does not allow a user to regen the kernel and move it into place without rebooting the system.

X11 Window System (See X Graphical User Environment)

X11 Windows Dev. Toolkit (See X Graphical User Environment)

X Graphical User Environment

HP-UX 9.0 contains X11R5, Motif 1.2 and VUE 3.0.

X11 R5 Server

X11 Server

The X11 server for the 9.0 release conforms to the X11 R5 release of the X Window System from the X Consortium.

The X11 R5 server is only supported on HP 9000 Series 300, 400, and 700 computer systems. The X11 R5 server is not supported on HP9000 Series 800 systems. The Series 800 HP-UX 8.0 X11 R4 server is not supported on HP-UX 9.0. Therefore, HP 9000 Series 800 users who require an X11 server cannot update to HP-UX 9.0.

X11 Server: Compatibility with previous releases

The X11 R5 server is fully backwards compatible at the X protocol level with the R4 servers in previous releases. Most existing X11 clients (R4 or earlier) that operate properly with the HP-UX 8.0 or 8.0* X11 R4 servers should operate properly with the HP-UX 9.0 X11 R5 server. A few old clients, however, may rely on defects in the R4 (or earlier) servers which have been fixed in the R5 implementation. See the **Bug Compatibility Mode** section below for a potential workaround.

X11 Server: New graphics device support

The HP-UX 9.0 X11 R5 server has support for the new A2091A CRX-48Z graphics device.

X11 Server: Bug compatibility mode

Like previous X11 servers, the X11 R5 server supports a bug compatibility mode of operation. When bug compatibility is enabled, certain innocuous defects which were in previous releases of the X11 server are re-enabled to permit old X clients which depend on the defects to operate with the R5 server. The default state for the HP-UX 9.0 R5 server is for bug compatibility mode to be disabled.

Bug compatibility mode can be enabled or disabled by the user in two ways. If the X server is invoked with the command line option `bc`, bug compatibility mode is enabled. Alternatively, the `xset` client can be used to toggle the state of bug compatibility mode:

```
xset [-display host:dpy] -bc    (disables bug compatibility mode)
```

```
xset [-display host:dpy] bc     (enables bug compatibility mode)
```

X11 Server: Serial input device support

The X11 R5 server has the new capability to support serial input devices (keyboard or pointing devices) through the dynamic loading of device drivers for these devices at run-time. Refer to the file:

```
/usr/contrib/X11drivers/input/serialdrv.doc
```

for more information on this facility.

X11 Server: Keyboard keymaps

The HP-UX 9.0 X11 R5 server keymap file (`XHPKeymaps`) now includes keymaps for many new localized Enhanced Vectra HIL keyboards.

X11 Server: New rgb.txt color name database files

Two new `rgb.txt` color name-to-RGB value database files are included with the HP-UX 9.0 X11 R5 release. These files contain several new R5 color names.

Also, the RGB values in the files have been customized for particular HP display model product series.

The files can be found in:

```
/etc/newconfig/X11R5/rgb.txt.98xxx
```

and

```
/etc/newconfig/X11R5/rgb.txt.A10xx
```

The `rgb.txt.98xxx` file was customized for color appearance on HP displays with product numbers beginning with 98.

The `rgb.txt.A10xx` file was customized for color appearance on HP displays with product numbers beginning with A10.

XhplibX11 XhpWindowToFile(), XhpFileToWindow(),
 XhpPixmapToFile(), XhpFiletoPixmap(),
 XhpQueryImageFile() Library Functions

Support for the following HP-proprietary functions is being phased out beginning with the HP-UX 9.0 X11 release.

```
XHPWindowToFile()  
XHPPixmapToFile()  
XHPFileToWindow()  
XHPFiletoPixmap()  
XHPQueryImageFile()
```

Documentation for these functions is being removed from the HP-UX 9.0 learning product materials. Code for the functions are removed from `libXhp11` at the next major HP-UX release.

X11 R5 Xlib (libX11): Device Independent Color Support

The new X Color Management System (`Xcms`) functions in R5 Xlib (`libX11`) support device-independent color spaces derivable from the CIE XYZ color space. This includes the CIE XYZ, xyY, L*u*v*, and L*a*b* color spaces as well as the TekHVC color space. In addition, linear RGB intensity value space was added as well as gamma correction for device RGB values. A uniform syntax was adopted for specifying colors in strings.

Xlib now supports client-side color name databases, and the existing functions that use color names (for example, **XLookupColor** and **XAllocNamedColor**) now handle all color spaces so that the contrivance of using **XParseColor** followed by **XAllocColor** is no longer necessary. **Xlib** now provides direct programming interfaces for dealing with color values in different spaces, and for converting between spaces. Also, new device-independent color spaces can be added.

The device-independent color model greatly expands the scope of visual attributes and specifications available to **Xlib** programs while maintaining compatibility with all existing **Xlib** programs.

Two sample device color characterization (**.dcc**) files are included with the HP-UX 9.0 X11 R5 release. These files can be used with the new **xcmsdb** utility, which provides a mechanism for placing screen color characterization data into root window properties. The files may be found in:

`/etc/newconfig/X11R5/98xxx.dcc`

and

`/etc/newconfig/X11R5/A10xx.dcc`

The **98xxx.dcc** file was customized for color appearance on HP displays with product numbers beginning with **98**.

The **A10xx.dcc** file was customized for color appearance on HP displays with product numbers beginning with **A10**.

Series 800 ONLY

An X11 R5 display server is not delivered or supported for the HP-UX 9.0 S800 release. The HP-UX 8.0 display server is not supported on HP-UX 9.0. Current users running HP-UX 8.0 who want to continue to utilize an X11 display server must not update to 9.0.

X11 R5 Font Server

Features

The font server is new for the X11 R5 release and HP-UX 9.0. It is in a new fileset called **X11-FONTSRV**. The font server provides networked font services to X11 display servers (either on workstations or X terminals) and

X11 clients. Network font services include access to bitmap fonts, generation of bitmap fonts from scalable fonts, and management of font licensing. The HP-UX 9.0 font server includes scalable font rasterizers for Intellifont (Agfa/Compugraphic) and Type 1 (Adobe) scalable fonts, as well as a limited bitmap font scaling capability. Several sample Intellifont and Type 1 scalable outlines are included with the font server in the HP-UX 9.0 release.

Running the Font Server

The font server process is not automatically started on your system. Whether or not you need to start the font server depends on your font requirements and your network configuration (e.g. Is there another font server already available on the network for your X server to access?).

The font server may be started on a system with the command:

```
/usr/bin/X11/fs -daemon
```

If the font server is to be run on the same machine as the X server and if the font server is started before the X server, the font server does not need to be added to the font path of the X server. However, if the font server is started after the X server is already running, or if a font server is running on a different machine than the X server, then the font server must be added to the font path of the X server as follows:

```
xset +fp tcp/:7000
```

This adds the font server port to the front of the font path list searched by the X server. If you are adding a font server that is not running on the same machine as the X server, the **xset** command is extended to also specify the font server host:

```
xset +fp tcp/hostname:7000
```

Once the font server is running and has been added to the font path of an X server, the server must **rehash** its fonts. This is also done with the **xset** client via:

```
xset fp rehash
```

Summary of Change

With HP-UX 8.0, Intellifont scalable outlines were delivered in the X11-FONTA fileset. With HP-UX 9.0, all scalable font outlines (Intellifont and Type 1) are delivered in the new X11-FONTSRV fileset. The X11-FONTA fileset still contains HP Roman8 bitmap fonts.

The X11 R5 release also has a new font format. X11 bitmap fonts are now delivered in **pcf** format instead of **snf**. A new utility, **bdf2pcf**, replaces the old **bdf2snf** utility for converting fonts from standard **bdf** format to **pcf** format. All fonts are shipped in compressed form, and font file names end with a **.pcf.Z** suffix).

All existing TypeDirector/UX utilities are still supported on HP-UX 9.0.

Alternatives and Compatibility

The X11 R5 font server accommodates bitmap fonts in the new **pcf** format, the old **snf** format, and standard **bdf** interchange format. A new utility, **bdf2pcf**, is included to convert fonts from **bdf** to **pcf** format.

All existing TypeDirector/UX utilities are still supported on HP-UX 9.0.

X11R5 Clients

Features

This release updates the X client's executables to MIT's X11 R5 standard. Since MIT is downgrading their support level for some clients, and has added other new ones, there are some changes to the population of clients that we are offering in this release.

Overall, the clients offered contain the general new features that come with X11 R5 and are mentioned elsewhere in this report, namely:

- Application resources can now be set on a per-screen basis and resource files can now contain ? wildcard characters and can include references to other resource files.
- Internationalization of the clients uses the new comprehensive R5 standard for internationalizing X clients. Clients that were internationalized with 8.*

are internationalized at 9.0, but those that were not, such as **xterm**, are not internationalized in 9.0.

- Perception-based device-independent color management as implemented in the R5 libraries is supported, which involves a new client that handles the color management database.
- Scalable fonts and font server protocol, to the degree that MIT clients take advantage of this.
- A new user authorization scheme was implemented for R5 as a step toward secure X. HP provides the clients that are necessary to support this scheme.

Hpterm now supports the use of up to 24 softkeys; see the descriptions of the **softkeyInitializeAll** and **softkeyLast** entries in the HP-UX Reference. The **dc1Handshake** resource enables the user to select how DC1 handshaking is controlled. The **roman8** resource gives the user the ability to control whether **hpterm** uses **roman8** encoding, or ISO-88591 by default. Support for Hebrew, Turkish, and Greek languages was added.

Two different flavors of **hpterm** are delivered with this release. The one built on X11R4 versions of **Xlib**, and **Xt** and Motif 1.1 is delivered by default. The version built on X11R5 versions of **Xlib**, and **Xt** and Motif 1.2 is delivered in a compressed shar file **/usr/bin/X11/hpterm.R5.sh.Z**; to install this version you must uncompress and unshar **hpterm.R5.sh.Z**. See **README.hpterm** in **hpterm.R5.sh.Z** for more information on how to choose between the R4-based **hpterm** and the R5-based **hpterm**.

The **xterm** terminal emulator has new features:

- Cuts of wrapped lines are now treated as a single line.
- Cuts of multi-page regions now work and highlight correctly.
- Pasting of large amounts of data into **xterm** now works.
- Various new resources have been added.

Summary of Change

The following clients are new to our product release:

xcmsdb Manages screen color characterization data.

The following clients were supported in HP-UX 8.*, but are not included in 9.0.

3-130 Topics Related to the HP-UX System

<code>xinit</code>	This method of starting the X server was downgraded in favor of the <code>xdm</code> startup method, which also supports X terminals.
<code>x11start</code>	This HP-proprietary startup script is closely associated with <code>xinit</code> and is likewise not included in the 9.0 release.
<code>xrefresh</code>	A screen refresher made obsolete by the standard refresh functions of the VUE and Motif window managers.
<code>xwininfo</code>	A X window info utility having low support and usage.
<code>xcutsel</code>	Old-style cut selection method that is now obsolete.
<code>bdftosnf</code>	A R4 font conversion program; obsoleted in R5 by <code>bdftopcf</code> due to a generalization in the format of font files.
<code>xfc</code>	An old R3 font conversion program symbolically linked to <code>bdftosnf</code> in R4 for 8.0, which is now itself obsolete.
<code>xfd</code>	X Font displayer; low level of customer usage.

Impact

The primary impact is to drop support for the `xinit` or `x11start` Xstartup method. Instead the program, `Vuelogin`, which is directly based on MIT's startup mechanism, `xdm`, is used. `Vuelogin` is not tied to the VUE environment (and `xdm` is not) and can be used to start up a non-VUE standard X session without using `xinit`. The main difference between the `xinit` startup method and the `xdm` or `vuelogin` method is that the former starts the X server, whereas the latter operates with the X server pre-started automatically. This is consistent with X terminals. There are some differences in the startup files associated with these two methods. Switching to the standard MIT startup method is noticed primarily by the customers who have not been using HP VUE or `xdm` under 8.*.

Performance

The X11 R5 release offers performance gains, primarily resulting from the use of the R5 libraries. The increased size of the `Xlib` library, due to the additional localization features, causes that shared library to grow.

Alternatives and Compatibility

Customers impacted by the loss of support for some clients that were supported in HP-UX 8.* should get the X11 R5 contrib distribution from the Interwork User Group.

X11R5 Xlib

Features

New functionality was added to **Xlib**: device independent color, and internationalization (**i18n**).

Summary of Change

The **Xcms** (X Color Management System) functions in **Xlib** support device-independent color spaces derivable from the CIE XYZ color space. This includes the CIE XYZ, xyY, $L^*u^*v^*$, and $L^*a^*b^*$ color spaces as well as the TekHVC color space. In addition, linear RGB intensity value space was added as well as gamma correction for device RGB values. A uniform syntax was adopted for specifying colors in strings.

Xlib now supports client-side color name databases, and the existing **Xlib** functions that use color names (for example, **XLookupColor** and **XAllocNamedColor**) now handle all color spaces so that the contrivance of using **XParseColor** followed by **XAllocColor** is no longer necessary. **Xlib** provides direct programming interfaces for dealing with color values in different spaces, and for converting between spaces.

New device-independent color spaces can be added. Internationalization, an internationalized application, is adaptable to the requirements of different native languages, local customs, and character string encodings. The process of adapting the operation to a particular native language, local custom, or string encoding is called localization. A goal of internationalization is to permit localization without program source modifications or recompilation.

Internationalization

Internationalization in **Xlib** is based on the concept of a **locale**. A locale defines the **localized** behavior of a program at run-time. Locales affect **Xlib** in the following ways:

- Encoding and processing of input method text.
- Encoding of resource files and values.
- Encoding and imaging of text strings.
- Encoding and decoding for inter-client text communication.

Xlib provides support for localized text imaging and text input. Sets of functions are provided for multibyte (**char&***) text as well as wide character (**wchar_t**) text in the form supported by the host C language environment.

Controlling Keyboard Input Using HP's X Window System

The X Window System uses the concept of **keysyms** to control the mapping of keys into characters. The set of **keysyms** for a particular keyboard is organized into a table called the **keymap**. To get information about keyboard mapping or to set the keyboard mapping use the **xmodmap** command.

Mapping keyboard for both Extend-char and Meta

A common problem reported by people using HP's X Window System is the conflict between the use of the **extend-char** key to access the extended characters of **Roman8** or **Latin1** with HP's keyboards and the use of the **extend-char** key as a Meta key.

The default mapping is that both keys serve both purposes. However, with HP-UX 9.0 it is possible to configure the keyboard so that one key is used as the **extend-char** key and the other as the Meta key.

The **xmodmap** command can be used to inquire and set the mapping for keys on the keyboard. Run the following command.

```
xmodmap -pm
```

For a US or West European keyboard in the default state this will print.

```
xmodmap: Up to 3 keys per modifier, (keycodes in parentheses):
```

```
shift    Shift_R (0xc),  Shift_L (0xd)
lock     Caps_Lock (0x37)
control  Control_L (0xe)
mod1     Meta_R (0xa),  Meta_L (0xb),  Mode_switch (0x36)
```

```
mod2
mod3
mod4
mod5
```

The `mod1` modifier has entries for both `Meta` `keysyms` and for `Mode_switch` as well; and this creates a problem. The solution is to use `mod2` for `Mode_switch` and change the `Meta_L` key into the `Mode_switch` key. To do this, use `xmodmap` and execute the following command:

```
xmodmap mods
```

where `mods` contains the following four lines:

```
remove Mod1 = Meta_L Mode_switch
keysym Mode_switch = NoSymbol
keysym Meta_L = Mode_switch
add Mod2 = Mode_switch
```

The entries in the file need to be in this order. Again, type:

```
xmodmap -pm
```

The results should be:

```
xmodmap: Up to 3 keys per modifier, (keycodes in parentheses):
```

```
shift    Shift_R (0xc),  Shift_L (0xd)
lock      Caps_Lock (0x37)
control   Control_L (0xe)
mod1      Meta_R (0xa)
mod2      Mode_switch (0xb)
mod3
mod4
mod5
```

The keyboard then uses the left `extend-char` key for extended characters and uses the right `extend-char` key for `Meta`. The client must be linked against `R4` or `R5 Xlib` for this to work.

Dead Key Compose processing

HP's X Window System has supported dead key compose processing for HP workstations for some time. This capability is now supported for non-HP servers (workstations and X-terminals) connected to HP systems.

In this form of compose processing a mute (or dead) key is struck followed by a second key. The initial key is a diacritic and the second key is the ASCII character to which the diacritic is to be applied. The diacritic character must be a special muting **keysym** to initiate the dead-key compose processing. The list of **keysym** names and the diacritic character to which they apply follows.

keysym name	diacritic character
hpmute_acute	´
hpmute_grave	`
hpmute_asciicircum	^
hpmute_diaeresis	¨
hpmute_asciitilde	~
acute	´
diaeresis	¨

To find out which muting diacritics are supported by a keyboard type:

```
xmodmap -pk
```

The entries in the third and fourth column of the keymap are the **extend** and **shift-extend** characters.

To set the mute keysyms as they are for HP series 700 terminals execute:

```
xmodmap mutes
```

where **mutes** is a file containing the following five lines.

```
keysym r = r R hpmute_acute
keysym t = t T hpmute_grave
keysym y = y Y hpmute_asciicircum
keysym u = u U hpmute_diaeresis
keysym i = i I hpmute_asciitilde
```

This is the default condition for HP's ITF keyboards.

Multi-key Compose processing

For release 9.0, HP's X Window System supports a form of compose processing that can be done using only ASCII characters. To use this form of compose processing, set a **keySYM** to the Multi-key **keySYM**. For example the **Enter/Print** key on an ITF keyboard could be used as the Multi-key. To do this, execute the following command:

```
xmodmap -e keySYM Execute = Multi_key
```

Then, compose processing can be done by typing the Multi_key (**Print**) followed by two other keys. One key should be the ASCII key that corresponds to one of the diacritic symbols and the other key should be the ASCII character to which the diacritic should be applied. The two keys can be typed in any order. For example, typing **Print ' e** generates a null character. The following table shows the ASCII characters and their diacritics.

ASCII character	diacritic character
,	˘
‘	˙
^	ˆ
:	˝
	˜
~	˘

Impact

The device-independent color model greatly expands the scope of visual attributes and specifications available to **Xlib** programs.

The internationalization support provides developers with a industry-standard method of creating multi-locale applications.

Performance

Databases should be significantly smaller in memory, and loading and parsing resources should be faster. Clients that are built archived see an increase in code size due to the support for device-independent color and internationalization.

Alternatives and Compatibility

The device-independent color model is compatible with all existing **Xlib** programs. The internationalization support does not affect existing applications.

Obsolescence

Fortran bindings are not shipped with HP-UX 9.0. The support for the bindings is now delivered with the Fortran language.

X11R5 Input Extension

There is an error in the **XInput.h** header file. User programs that include **XInput.h** and compile with the **-Aa** option gets a compilation error. The problem is due to ANSI prototypes which were added to **XInput.h** for the 9.0 release.

Two workarounds exist for the problem:

1. Add the following lines to the application program ahead of the include for **XInput.h**:

```
#define NEED_EVENTS
#include X11/Xproto.h
```
2. Delete the ANSI prototype for **XInputEventToWire** from **XInput.h**. (This routine is called by **XSendExtensionEvent**. Clients should never call it directly. The ANSI prototype should therefore never be needed and can therefore be removed).

X11R5 Xt

Features

Most of the major changes made to the Intrinsics relate to improved performance and customization. At the data-structure level, Release 5 retains complete binary compatibility with Release 4.

Summary of Change

The Release 5 Intrinsics have implemented memory-saving measures that include the reorganization of string constants into a single array, a rewrite of the translation manager, and the use of permanently allocated strings where possible.

Base Translations Resource

A new pseudo-resource, `XtNbaseTranslations`, permits application developers to specify translation tables in application defaults files while still giving end users the ability to augment or override individual event sequences. This change affects only the applications that wish to take advantage of the new functionality or the widgets that may have previously defined a resource named `baseTranslations`.

Resource File Search Path

The current specification allows implementations greater flexibility in defining the directory structure used to hold the application class and per-user application defaults files. Previous specifications required the substitution strings to appear in the default path in a certain order, preventing sites from collecting all the files for a specific application together in one directory. The Release 5 specification allows the default path to specify the substitution strings in any order within a single path entry. Users need to pay close attention to the documentation for the specific implementation to know where to find these files and how to specify their own `XFILESEARCHPATH` and `XUSERFILESEARCHPATH` values when overriding the system defaults.

Customization Resource

`XtResolvePathname` supports a new substitution string, `%C`, for specifying separate application class resource files according to arbitrary user-specified categories. This provides separate monochrome and color application class defaults files. The substitution value is obtained by querying the current resource database for the application resource name, `customization`, and class, `Customization`. Any application that previously used this resource name and class needs to be aware of the possibly conflicting semantics.

Per-Screen Resource Database

To allow a user to specify separate preferences for each screen of a display, a per-screen resource specification string was added, and multiple resource databases were created; one for each screen. This affects any application that modified the (formerly unique) resource database associated with the display subsequent to the Intrinsics database initialization. Such applications need to be aware of the particular screen on which each shell widget is to be created.

Impact

There should be no significant impact on developers from the R5 Intrinsics.

Performance

Most applications should see a decrease in memory usage. In a few cases, execution time should be faster. Clients that are built archived see an increase in code size.

Alternatives and Compatibility

The R5 Intrinsics are binary and API-compatible with the R4 Intrinsics.

HP Motif 1.2

Features

The OSF/Motif 1.2 release contains new R5-based internationalization capability, drag-and-drop data interchange, tear-off menus, CUA Style Guide compliance changes, general enhancements, and improved performance relative to the Motif 1.1 release. Minor enhancements were made in `mwm` and `UIL`.

Summary of Change

Many of the changes to Motif for Revision 1.2 were made to enable the development of internationalized applications. Most dramatic were the changes to the `Text` and the `TextField` widgets, including interfaces that support both fixed-width wide characters (`wchar_t`) and variable-width multibyte characters (`char *`). The Compound String routines were modified to handle locale-encoded strings and to integrate X11 R5 fontset capability.

X11R5 uses the concept of an input method (IM) that provides a layer of mapping between the keys typed by users and the corresponding characters passed to an application. The Toolkit was enhanced to support the transparent connection to an input method for those languages that require an input method. The **VendorShell** widget was modified to support the geometry management of an Under-the-Spot input method.

The **XmDisplay** widget was added to support internationalization. The **XmScreen** widget was added to support the specifications of resources on a per-screen basis.

Drag and Drop Capability

In compliance with the OSF/Motif Style Guide, Revision 1.2 supports the drag and drop metaphor for data interchange. The Drag and Drop specification was fully implemented. Refer to the OSF/Motif Programmer's Guide and the HP-UX Reference for information on the drag and drop interface.

Tear-off Menus

When Tear-off Menus are enabled by the application, the user can retain menus on the display for subsequent selections. Each tearable menupane has a tearoff button. When the mouse drag button is pressed on the tearoff button, the pane tears off and can be dragged and then placed by releasing the mouse drag button. The window manager surrounds the tearoff menupane with a menu button and a title. Shifting focus to a tearoff windowpane follows standard window manager policy.

Performance

The **XmText** widget was modified to use a line table rather than a one-dimensional array. Interactive performance is improved relative to Motif 1.1.

Insensitive Visuals

Visual indications are provided to indicate whether a component can respond to input from users. Labels and buttons have had this behavior in previous releases. In Release 1.2, the behavior is extended to the **Text**, **TextField**, **ScrollBar**, and **ArrowButton** widgets.

Titles for Frames

In Revision 1.2, title labels can be displayed in a **Frame** widget. New resources were added for specifying the position of the title in the frame.

Audible Warning

A new resource, **XmNaudibleWarning**, was added to the **VendorShell** widget to specify whether an audible cue should accompany a warning message. **Text** widgets determine the value for this resource from the value of **XmNaudibleWarning**.

Color Enhancements

Users can specify default background color and thresholds for shadow calculation on a per screen basis with these 3 new resources on the **XmScreen** widget:

- **XmNlightThreshold**
- **XmNdarkThreshold**
- **XmNforegroundThreshold**

A new function, **XmChangeColor**, changes the background and other colors for a specified widget.

Baseline Alignment

A new function, **XmWidgetGetBaselines**, determines the position of the widget's text baseline. Another function, **XmWidgetGetDisplayRect**, determines the size and position of the character cell.

2-D Menu Traversal

The left, right, up, and down traversal arrows navigate within a menupane. The up and down arrows wrap between columns, and the right and left arrows post the previous or next menupane.

Input Focus

A new resource, `XmNInitialFocus`, was added to the `Manager` class to specify the first widget to receive input focus. This resource specifies a widget and is ignored for all popup menus, menubars, option menus, and pulldown menus.

Traversal Access Functions

To support better interaction with keyboard traversal, Revision 1.2 includes these new functions:

- `XmGetFocusWidget`
- `XmGetTabGroup`
- `XmIsTraversable`
- `XmIsVisible`
- `XmGetVisibility`

The `XmTrackingLocate` function includes the following modifications:

- The function fields all events, not just the button press.
- The function returns on any keystroke or a button press.
- The function can be called for non-sensitive widgets.

A new function, `XmTrackingEvent`, was added, which is just like `XmTrackingLocate`, except that it returns a pointer to the X event.

Virtual Keys

A new function, `XmTranslateKey`, provides a mechanism for applications to define their own `XtKeyProc`, which turns off virtual key processing. Also, a database is looked up in `/usr/lib/Xm/bindings`.

Resource Management

Several new functions were added for managing representation types, including:

- `XmRepTypeRegister`
- `XmRepTypeAddReverse`
- `XmRepTypeValidValue`

- XmRepTypeGetRegistered
- XmRepTypeGetId
- XmRepTypeGetNameList
- XmRepTypeGetRecord

Changes for CUA Windows Compliance

In Motif Release 1.2, **KActivate** only activates the default action in Menus and DialogBoxes. In **XmBulletinBoard**, only action buttons (**PushButtons** and subclasses) can be the default when the user presses **KActivate**. Traversing from one action button to another makes the currently selected button become the default.

Specific Widget Changes and Enhancements

Several minor enhancements were made to the following widgets:

- XmBulletinBoard
- XmCommand
- XmList
- XmScrolledWindow
- XmRowColumn
- XmScrollBar
- XmSelectionBox
- XmText
- XmTextField

Refer to the Motif documentation for descriptions of specific changes.

MWM Enhancements

The following changes were incorporated into the Motif Window Manager (**mwm**):

- An internationalized **.mwmrc** file.
- Support for the **Return** continuation character in the **.mwmrc** file.

- Search `XBMLANGPATH` for bitmap files.
- Support for tight icon placement.
- Support for pop-down and replay event behavior in `mwm` menus.
- Support scrolled window traversal to scrolled-off children in icon box.
- Support for geometry of the feedback window with the geometry resource.
- Support for the `SHAPE` non-rectangular window extension.

The window manager includes these new resources:

<code>stackGrouping</code>	Determines whether a window family should be stacked as a group with its parent window.
<code>transientStackingPolicy</code>	Determines whether all application windows are lowered when a secondary window is lowered.
<code>moveOpaque</code>	Controls whether an image of the window, or just an outline, is moved.

The window manager includes these new and enhanced functions:

<code>f.restore</code>	Restores a window to its previous state. Double clicking on an icon is bound to this function rather than to <code>f.normalize</code> .
<code>f.lower</code>	Includes a <code>within</code> argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a <code>freeFamily</code> argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.
<code>f.raise</code>	Includes a <code>within</code> argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a <code>freeFamily</code> argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

f.raise_lower Includes a **within** argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a **freeFamily** argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

There is one new **mwm** action: **Alt Esc**

It behaves similar to the **f.next_key**, except that the window is always raised regardless of the value of **focusAutoRaise**.

An **mwm** puts a process spawned with **f.exec** in its own process group. An **xterm** or **hpterm** started with **f.exec** does not inherit the tty. The settings of **mwm**'s controlling terminal use the **ttyModes** resource to define tty settings for your terminal emulators.

UIL Changes

The following changes were made to UIL:

- A new command flag, **-s**, was added to enable the use of **setlocale**.
- New UIL syntax to support font sets and font tables.
- New UIL syntax to support wide character strings.
- Syntax changes to WML to allow for the definition of automatically created children of composite widgets.
- UIL can read binary databases (WMD files) containing WML information with the **wmd** option.
- **Mrm** includes one new function, **MrmOpenHierarchyPerDisplay**, which is the same as **MrmOpenHierarchy**, except that **display** is passed as an explicit argument. This function replaces **MrmOpenHierarchy**.

UIL Installation Note

Installation of the X11R5 Developer's Environment destructively replaces the Motif 1.1 UIL compiler with the Motif 1.2 UIL compiler. The Motif 1.2 UIL compiler cannot be used to compile UIL source files for Motif 1.1 applications. If additional UIL development for Motif 1.1 applications is expected, then the

Motif 1.1 UIL compiler (`/usr/bin/X11/uil`) should be given another name before proceeding with the installation of the X11R5 Developer's Environment.

Impact

Applications should be able to relink or recompile with no problem. In addition, numerous improvements were made in the quality of the widgets, so that applications should benefit from this as well.

Performance

Interactive text performance is much improved, especially in the area of scrolling. Memory-saving changes were implemented, but some of the savings were offset by space taken by new functionality. Most applications should see a minor decrease in memory usage. Clients that are built archived see an increase in code size.

Alternatives/compatibility

Motif 1.2 is source and binary compatible with Motif 1.1. However, there are a few minor changes that developers should note.

Include File Changes

In the Motif 1.1 release, links to the `AtomMgr.h`, `Protocols.h`, `ProtocolsP.h`, and `MwmUtil.h` header files were placed in `X11` and `Xm` in the `make includes` and `make install` procedures. This was for backwards compatibility with the 1.0 release. The `make includes` and `make install` procedures no longer create these links. The files should be included as `Xm/ ...` instead of `X11/ ...`.

The following header files were in Release 1.1 of Motif, but are now obsolete in Release 1.2:

- `ExtObject.h`
- `Traversal.h`
- `VaSimple.h`
- `VendorE.h`
- `VendorEP.h`

The files are maintained for source compatibility.

Other Visual Changes

The following items describe minor visual differences with respect to Motif 1.1:

- Several pixels have changed in the 3-D beveled look.
- Revision 1.2 color generation routines include minor changes.
- There are minor layout differences due to bug fixes in geometry management.

File Set Name Change

X11-SHLIBS from the previous release of HP-UX was renamed X11R4-SHLIBS. Update issues a warning if you try to load an application that depends on X11-SHLIBS. The application still works if you load X11R4-SHLIBS from the OS media.

HP VUE 3.0

VUE 3.0 runs on 9.0 systems. There are differences between VUE 2.01, which ran on previous releases, and VUE 3.0 in the areas of installation, product structure, and component characteristics. The following sections list the differences.

Installation

VUE 2.01 files are installed in many places throughout the file system. VUE 3.0 files are mostly installed in the `/usr/vue` branch of the file system. This includes, `app-defaults` files, icons, executables, and configuration files. Help files are installed in the `/usr/vhelp` directory, facilitating easier removal in a help file server configuration.

Dr_Vue: HP VUE Diagnostic Tool

Dr_Vue is an HP VUE 3.0 diagnostic tool located in the `/usr/contrib/bin/X11/` directory. The purpose of `Dr_Vue` is twofold:

1. To check on the status of the HP VUE fileset, and
2. To check on the runtime configuration of the system.

Dr_Vue does not initiate any fixes directly. Instead, Dr_Vue diagnoses and reports problems.

If HP VUE is currently running, **Dr_Vue** is initiated by selecting the **General** toolbox inside the Toolboxes subpanel and double clicking on the **Unsupported** toolbox. Once inside that toolbox, double click on the **Dr_Vue** icon. An action associated with the **Dr_Vue** icon creates an **hpterm** window and runs **Dr_Vue**.

Dr_Vue can also be run from the command line by typing the following:

```
/usr/contrib/bin/X11/dr_vue | more
```

Dr_Vue issues INFO, WARNING, or ERROR statements with information that should help expose problems that exist with the installation or configuration of HP VUE. There is one WARNING statement that should be ignored; it states that the **/usr/vue/nls** directory is missing. This directory should not be present.

Backward Compatibility

VUE 3.0 is functionally compatible with VUE 2.01. User customizations of VUE that affect functionality are supported (e.g., action customizations, controls added to the front panel). Visual customizations are generally supported, but, given the extensive new VUE 3.0 visuals, often have to be redone to be attractive.

This summary addresses the impact of moving from HP VUE 2.01 to 3.0. Examples of backwards compatibility include:

- Workspace Manager reads the user's 2.01 front panel configuration if one exists; otherwise, the new 3.0 front panel is used.
- A 2.01 to 3.0 front panel conversion program (**vuefp2to3**) is available.
- VUE 2.01 help is preserved and is available from the front panel slideup along with the new 3.0 help volumes.
- Customized 2.01 action and filetype specifications are used
- A 2.01 to 3.0 action/filetype conversion script is available.
- The location of 3.0 toolboxes has changed from 2.01. The 2.01 toolboxes are available within the 3.0 toolboxes.
- 2.01 display dependent sessions are used. Online step-by-step help exists which explains how to move to display independent sessions.

- The **Moving To HP VUE 3.0 from HP VUE 2.01** online help is available which provides detailed information and step-by-step instructions.

The end user's 2.01 environment is carried across to 3.0. User noticeable affects of updating are described below.

System Configuration Files

Most HP VUE 3.0 files are installed in the `/usr/vue` directory instead of scattered throughout the file system as they were in 2.01. The ramifications of this change on system configuration files are described below.

The set of configuration files used by Login Manager and Session Manager to provide default system startup are customizable by system administrators. These files are moved to `/usr/vue/VUE2.0config`. Therefore, customizations to these files are temporarily lost until the system administrator merges the customizations with the new default configuration files. These include X server startup (`Xservers`), the appearance of the login screen (`Xresources`), the system default front panel (`sys.vuewmrc`), the system default session (`sys.session`), the default session startup script (`Xsession`), `Xstartup`, `Xreset` and `Xconfig`.

The impact varies depending on the extent of system level customization. The auto-customization portion of the `Xsession` script is maintained from 2.01 to 3.0; it is copied into the 3.0 `Xsession` script.

The 3.0 installation process removes files delivered with 2.01, including, VUE specific directories (as long as they contain only files delivered with 2.01). Files not delivered with 2.01 that are in VUE specific directories are not removed.

Mixed Networks

Interoperability problems exist with the File Manager for networks running both 2.01 and 3.0 with nfs-mounted home directories. In general, problems can occur when a user is manipulating their single home directory concurrently from different workstations.

User Sessions

The user's 2.01 session is moved to 3.0 intact with two exceptions. Hpterminals to remote systems running 3.0 come up, but in the current workspace rather

than the workspace specified. This is due to **hpterm**'s change in location from `/usr/bin/X11` to `/usr/vue/bin` and session information containing path information. To recover, users should move their **hpterm** to the correct workspace and save their session.

2.01 users logging into 3.0 the first time see the default 3.0 palette. Any palettes the user has customized are kept intact; the session palette setting is lost. To recover, the user must select the palette and save their session.

Backdrops

A few of the backdrops were obsoleted to save disk space. There is a compressed archive of these obsoleted backdrops and step-by-step online help instructions for recovering them (`/usr/vue/examples/vuebackd.sh.Z`).

Resources

System level customizations to the app-defaults files are lost. This is standard procedure for applications updating app-defaults files. Note that user level resources are preserved.

Front Panel

A few front panel resources have been obsoleted. These resources are for specifying geometries for the front panel itself, and the workspace switch control. This is handled automatically in 3.0. If specified, these resources have no effect.

If there is a customized 2.01 front panel specified in the `.vue` directory it is used with new 3.0 visuals (**vuewm** automatically runs the **vuefp2to3** conversion utility). The 2.01 front panel specification is left in place. To change the 2.01 specification to a 3.0 specification the **vuefp2to3** utility should be used. To get new 3.0 functionality the customized 2.01 specification should be removed or parts of the **system.vuewmrc** specification merged in (from `/usr/vue/config/system.vuewmrc`). Customized bitmaps referenced by the front panel specification are displayed with two colors and without a mask. The 3.0 color icon editor can be used to convert the bitmap to a multicolored, masked icon (use the transparent color to indicate the mask).

Front Panel Mail Indication

The mail indicator on the front panel may periodically revert from the **new mail** state with **elm** running. If **elm** is running when new mail arrives, the mail control indicates new mail for a period of time, then reverts to the normal mail icon. This is because **elm** periodically resynchronizes (wakes up to read the mail file), and then the mail control notices that the mail file has been read. To turn off resynchronization in **elm**, set **timeout** to zero in the **elmr**c file (**\$HOME/.elm/elmr**c).

For correct behavior, mail controls in the front panel must be of type **mail**, not **monitor_file**. The front panel converter may insert the wrong control type when converting a custom VUE 2.0 front panel if the name of the control is not **Mail**.

Window Management

Some users may notice new window or workspace behavior on some applications running in VUE 3.0. The new behavior you might see is that a window occupies more than one workspace over time as you move around between workspaces.

This change is part of a defect fix to allow windows that are not in the current workspace to get into the current workspace by simply mapping themselves again. The old behavior of **vuewm** broke some applications by confining some of their dialogs to other workspaces and fragmenting their user interface.

AbsentMapBehavior is a client-specific resource, so this behavior can be easily tuned. It is documented in the **vuewm(1X)** entry in the HP-UX Reference.

VUE Lite

There are two primary configurations of VUE: 1) full VUE, and 2) VUE Lite. Users can select (on a display by display basis) between full VUE and VUE Lite sessions. VUE Lite is a reduced version of VUE that has performance characteristics similar to a non-VUE (**x11start** plus **mmm**) configuration.

VUE Lite includes:

- **vuelogin**
- **vuewm** (workspaces, front panel, action invocation)

- **vestyle** (with reduced functionality in color, font, background and window manager customization)

- rudimentary session management

VUE Lite does NOT include:

- **vuefile**
- drag and drop
- full **vestyle** customization (mouse settings, keyboard settings, etc.)
- **vesession** (save clients, mouse/keyboard settings, etc.)
- screen locking

General Appearance And Behavior Changes

- Motif Style Guide Compliance
- New help presentation and information for all components
- Multicolored icons
- Icon library

New VUE Components

- Graphical editor, **vuepad**.
- Color icon editor.
- File annotator (including text and audio annotations)

New Courtesy Clients

These clients are not tightly integrated into the VUE environment and are provided with minimal support.

- Financial calculator
- **xcolumns** game
- Action diagnostic tool

VUE Library

The VUE 2.01 library (**libXv.a**) continues to be shipped with VUE 3.0 to support existing non-VUE clients that used the VUE API (for on-line help access and session management). The on-line help API is in a separate library for VUE 3.0 (see On-Line Help below) and a new session management API is NOT being provided. ICCCM mechanisms are sufficient to provide application session management.

On-Line Help

A new help library and client is delivered with VUE 3.0. New features and characteristics of the library include:

- No dependencies on the VUE product.
- Works with Motif 1.1 and 1.2.
- Provides a standard look and feel interface for displaying, accessing and processing help information.
- Help information can incorporate multiple fonts and graphics.
- Hypertext and hypergraphics links.
- Standard Motif cut and paste.
- Localizable (including 16 bit languages).
- Help information can be printed (including graphics and 16 bit fonts).
- Authoring markup based on the SGML standard.

A new help browsing client (**helpview**) based on the new help library is delivered with VUE 3.0. The VUE 2.01 help client (**vuehelp**) is shipped with VUE 3.0 to handle non-VUE help text written with the VUE 2.01 **vuehelp** format.

Login Manager (vuelogin)

The Login Manager contains the following new features:

- System hostname displayed on login screen.
- Kerberos authentication is available in addition to standard authentication (subject to approval by government regulatory agencies).

- Customizable languages menu.
- VUE Lite/Full session can be selected at login time.
- The copyright screen after login has cosmetic changes.

Session Manager (vuesession)

The Session Manager manages and reduces contention between clients started up as part of a session (improves login performance on low memory systems). The Session Manager contains the following new functionality:

- Interactive selection of full screen cover when locking.
- A saved session can be used on more than one display.
- A saved session can more easily be transferred between systems.
- Optional locking of screen on timeout.
- Fonts selected on a per-language basis.
- Support for system-wide enabling of access to a display.
- Support for C2 secure system screen locking.

Window/Workspace Manager (vuewm)

The `vuewm` client was changed in two key areas:

1. general window manager functionality, and
2. front panel appearance, behavior and functionality.

General window manager changes include:

- `vuewm` and `vuesession` cooperate to improve login performance.
- Key resources for the window manager are easier to change (through a `vestyle` GUI)
- `vuewm` and `vestyle` communication performance enhancements.
- Help available through the [F1] key.

Front panel changes include:

- New visual design (`smooth` look).

- Support for multi-colored icons with masks.
- New multi-colored icons for the front panel controls.
- Slideup sub-panels.
- Support for animation on pushing a control.
- Support for front panel keyboard traversal.
- Clearer and more robust syntax for configuring the front panel.
- Cleanup of user changeable front panel configuration resources (more robust, consistent and manageable).

VUE 2.01 front panel specifications are automatically converted by the VUE 3.0 **vuewm**. Existing front panel customizations are functionally supported, but there may be some visual degradation.

Style Manager

The following are the new Style Manager features.

- Style management options can be constrained
- Easy selection of key window manager resources (keyboard focus policy, use of icon box, and so on).
- Easy selection of key session manager resources (screen lock on timeout).
- Support for static color displays.
- Easy configuration of VUE color use.
- Improved backdrop selection.

File Manager

Changes were made to the File Manager (**vuefile**) for VUE 3.0. These changes include:

- Use root window as a desktop for depositing frequently used objects for easy access.
- Multicolor icons.
- Click and type file renaming.

- Quick access to a terminal opened on the current working directory.
- Quick help on icons.
- Arbitrary placement of icons.
- Improved Find File and Filter File dialogs.
- Full support for keyboard traversal.
- Support for mouse button 1 drag and drop.
- Support for quick changes to an known directory.
- File content based file typing.

The VUE 2.01 Drag-and-Drop Manager was merged into File Manager process (for enhanced performance on startup and drag-and-drop interactions).

Action and File Type Databases

The action and file type database syntax was redesigned to be easier to specify and more robust. VUE 2.01 databases is automatically converted at run-time by VUE 3.0 components.

Command Invoker

The VUE 2.01 command invoker process is used to activate local and remote commands and applications. For VUE 3.0 the process was eliminated and turned into a library that is used by **vuewm** and **vuefile**. The cache of **hpterm**'s was eliminated (to improve login performance and cut down on system resource usage at the cost of run-time performance of actions that use **hpterm**s). The command invoker library also does automatic xhosting to allow remote command access to the user's display.

Toolbox

The organization of VUE toolboxes (application folders) was changed for VUE 3.0. Pushing the toolbox button on the front panel brings up the user's toolbox. The system toolbox and a catalog of applications (that the user can order) are accessible through the toolbox slideup.

Graphical Editor

The **vuepad** graphical editor is based on the Motif text edit widget and supports standard cut and paste, word wrap, drag-and-drop and printing. For VUE 3.0, **vuepad** is capable of handling 16-bit languages.

Color Icon Editor

An editor to create and modify colored icons is shipped with VUE 3.0. This editor can be used to produce icons that tie into VUE's color schemes and dynamic colors.

File Annotator

The file annotator is new for VUE 3.0 and allows the user to attach audio and text annotations to a file. It also can be used to view or file annotations.

Other Topics

This part contains information for topics related to specific products you purchase (for example, Starbase, C++, and NetLS).

ANSI/C Compiler

With the 9.0 release, many languages are available as separate products. As such, the products have their own release notes. The release notes for the products are placed on the system during the update process. They appear, after the update, in the following directory:

`/etc/newconfig/90RelNotes`

This situation applies to the ANSI/C Compiler.

C++

The 9.0 release includes a new release of C++ called HP C++ 3.0. The new release includes new enhancements as follows:

- Support for templates (parameterized types) and exception handling. HP C++ 3.0 is based on USL C++ 3.0.
- As a part of special bundling of C++, HP C++ also includes Standard Components, which is a class library of general purpose programming routines and tools.

The part numbers for HP C++ products are B2400A & B2401A (S400 HP-UX), B2402A & B2403A (S700) and B2404A & B2405A (S800). For more

information, see datasheet #5091-5375E, which is available from the Literature Distribution Center.

Features and Benefits of HP C++ 3.0

Feature	Benefit
C++ Compiler	Generates efficient code; Faster Compilation
Object Oriented Debug	Locate errors quicker;
Standard Components and other Class Libraries	Off the shelf software; Reduces code that needs to be written
Tools and Header files extended to work with C++	Enable use of existing software directly from C++
Templates or parameterized types	Promotes Software Reuse
Exception Handling	Provides a convenient mechanism for handling error conditions
Conforms to C++ ANSI base document	Standards compliant; Investment protection

GKS

The HP Graphical Kernel System (GKS) was removed from the Corporate Price List in March 1992. To continue HP's commitment to graphics standards HP has developed a partnership with Advanced Technology Center (ATC) to supply ATC-GKS. This product is marketed, supported, maintained, and packaged as an HP product.

HP C/HP-UX 9.0

The HP C product with the HP-UX 9.0 release includes significant enhancements to the compiler, tools, optimizer, and manuals. This section summarizes these new features.

New HP C Compiler Features

The following features have been added to the HP C Series 700/800 compiler:

- **Ae** command line option - Provides an extended ANSI mode that combines ANSI C with extensions. Combines **-Aa**, **-D_HPUX_SOURCE**, and enables the **sized enum** and **\$** in identifier extensions.
- **+DAmodel** command line option - Generates object code for a specific version of the PA-RISC architecture.
- **+DSmodel** command line option - Performs instruction scheduling for a specific implementation of PA-RISC.
- **+ESlit** command line option - Places string literals and constants into read-only data storage.
- **+ESsfc** command line option - Replaces function pointer comparison millicode calls with inline code.
- **+f** command line option - Inhibits the promotion of **float** data types to **double** data types when evaluating expressions. It performs promotions on arguments passed to functions and values returned from functions.
- **FLOAT_TRAPS_ON _ALL** pragma - Controls the optimization of floating-point operations when floating-point traps are enabled. Prevents loop invariant code motion on floating-point operations throughout the compilation unit. (See also **+OE**.)
- **+FPflags** command line option - Controls the initialization of the floating-point registers at run-time. Improves flexibility and control over floating-point exception handling. Can be used in place of the **HPENABLETRAP** HP-UX library routine.
- **HP_ALIGN** pragma enhancements - Controls the alignment of data items within structures, facilitating data exchange among various machine architectures.

- **-J** command line option - Generates inline assembly for **strcpy()**, **sqrt()**, and **fabs()**. The source library routines are expanded inline to provide faster execution. Note, using this option changes the error reporting behavior. The **matherr** function (3M) is not called and **errno** is not set. More routines may be added in future releases.
- **New Math Libraries** - Delivers math libraries that have been optimized for specific PA-RISC architectures.
- **+OE** command line option - Controls the optimization of floating-point operations when floating-point traps are enabled. Prevents loop invariant code motion on floating-point operations throughout the compilation unit. (Same as **FLOAT_TRAPS_ON _ALL.**)
- **+Om1** command line option - Enables the optimizer to assume the same FORTRAN parameter aliasing or memory overlapping rules as specified in ANSI X3.0-FORTRAN 77, page 15-20. This option attempts to insure that functions that have been literally translated from ANSI FORTRAN to C do not exhibit different behaviors as a result of differences in the way parameters are accessed.
- **+Os** command line option - Performs all standard level 2 optimization except those optimizations that might result in significant code size expansion. Software pipelining is disabled when this option is used.
- **Profile-Based Optimization** - A feedback based optimization that attempts to improve instruction cache, page, and Translation Lookaside Buffer (TLB) hit rates.
- **sized enum** data type - Controls the space allocated to enumerated variables.

For details about the Series 700/800 HP C/HP-UX 9.0 enhancements, see the HP C/HP-UX 9.0 Release Notes. The on-line release notes file is called `/etc/newconfig/90RelNotes/C`.

Network File System (NFS)

Features

At 9.0 NFS has the following NFS 4.1 (from Sun Microsystems) features:

- `exportfs(1M)`

- `export of root`
 - `directory exports`
 - `read-only exports`

- `automount(1M)`

- New options to the following commands:

- `mount(1M)`
 - `rpcinfo(1M)`
 - `ypbind(1M)` {NIS or Yellow Pages}

New Command: `exportfs(1M)`

The `exportfs(1M)` command provides the following new functionality:

- Export of Root

- Allows the NFS server to export `root` capability to NFS clients.

- Directory Exports

- Allows the NFS server to export individual directories and subdirectories to a client, instead of forcing exports to take place at the file system level only.

- Export Read-Only

- Allows the NFS server to export read-only capability on a file or directory to the client.

The `exportfs` command reads `/etc/exports` and manipulates the `/etc/xtab` file. It calls the system routine `exportfs` to keep the kernel up to date with the `/etc/xtab` file. The `exportfs` command uses the `libc/exportent` routines to manage the `/etc/xtab` file. See the `exportent(3)` HP-UX Reference entry for more information.

New Command: automount

The **automount(1M)** command allows NFS mounts to occur dynamically. When the client (user or application) attempts to access a remote NFS file, the file system mount is automatically and transparently established. After a period of inactivity (default is 5 minutes) **automount** unmounts the mounted file system.

Note that **automount** was available with 8.07 for the Series 700.

Enhanced Functionality: mount

The following options are new for the **mount(1M)** command:

noac	Suppress fresh client attributes when opening a file.
nocto	Suppress client attribute and name (lookup) caching.
acregmin=n	Hold cached attributes for at least n seconds after file modification.
acregmax=n	Hold cached attributes for no more than n seconds after file modification.
acdirmin=n	Hold cached attributes for at least n seconds after directory update.
acdirmax=n	Hold cached attributes for no more than n seconds after directory update.
actimeo=n	Set min and max times for regular files and directories to n seconds.

For more information, see the **mount(1M)** entry in the HP-UX Reference.

Enhanced Functionality for Several Commands

The **rpcinfo(1M)** command was updated to include a new option when listing registered RPC programs, a new option to discover any hosts running a specified program and version, and a new option to delete RPC registration for the specified program and version. See the **rpcinfo(1M)** entry in the HP-UX Reference.

The **rpc.mountd** command has a new option for tracing. See the **mountd(1M)** entry in the HP-UX Reference.

4-6 Other Topics

The `ypbind(1M)` command has a new option for fixing a security hole. This option is provided to allow backward compatibility for the `ypset(1M)` command. The `-ypset` option allows the `ypset(1M)` command to be executed to change the binding. For maximum security, this option should be used for debugging the network from a remote machine.

The `/bin/passwd` file and `ypmake(1M)` command now support NIS (YP) password aging for both local `/etc/passwd` files and NIS password maps. See Appendix A in the Installing and Administering NFS Services document.

Increase in NFS Limit

The NFS limit for user membership in groups has increased 8 groups to 16 groups (the HP-UX limit is 20). See the `group(4)` entry in the HP-UX Reference.

Any NIS operation accessing the host maps is now case insensitive. See `gethostbyname(3N)` in the HP-UX Reference.

Impact of New Features

exportfs(1M)

The only enhancement that requires attention for backward compatibility is the inclusion of the new `/usr/etc/exportfs` command. It is NECESSARY to run `exportfs` after changing the `/etc/exports` file directly.

Usage is: `/usr/etc/exportfs -a` (all)

It is necessary to run `/usr/etc/exportfs` with the `-a` option to have the new `/etc/exports` changes take affect. See the `exportfs(1M)` and `exports(4)` entries in the HP-UX Reference for more information.

The `/etc/netnfsrc` script was modified to execute `exportfs` at boot time. Therefore, administrators only need to run `exportfs` explicitly when there is a change to the `/etc/exports` file.

Only one entry per directory or file is valid in `/etc/exports`. In previous versions of NFS, the ability to have multiple entries for a file system in `/etc/exports` was an undocumented feature that no longer exists.

You can now export on a directory by directory basis. See the **exportfs(1M)** entry in the HP-UX Reference for more information.

You cannot export a directory that resides within the same file system and is either a parent or sub-directory of a directory that is currently exported. For example, **/usr** and **/usr/local** cannot both be exported if they reside in the same disk partition.

mountd(1M)

The **rpc.mountd** daemon should now be started from **/etc/netnfsrc**. Take the following actions:

1. Compare your **/etc/inetd.conf** file with the version in **/etc/newconfig**. Make edits as necessary and make sure the **rpc.mountd** entry is removed.
2. Edit the **/etc/netnfsrc** file by setting the **START_MOUNTD** variable to 1.

HP ARPA Services: gated(1M)

Features

The **gated(1M)** command incorporates three independent programs: Berkeley's **routed**, **egpup**, and **hello** programs. It also provides a common configuration file. The 2.1 version includes support for multiple timers and tasks to deal with events. It also provides an enhanced configuration-file format, making configuration and support easier. The following items note the 2.1 improvements for **gated(1M)**.

- Improved configuration.

The configuration file, **grammar**, was redefined and is now more consistent across all three protocols.

- Enhanced route control policy.

The route control policy was enhanced to provide more flexible route control. Specifically, the following features were added.

- Routes can now be listened to and/or propagated by protocol (for example, RIP), autonomous system, interface (for example `lan0`), source gateway.
- Preferences can now be used to order routes.
- Masks can now be used when specifying routes.

■ Border Gateway Protocol (BGP)

The Border Gateway Protocol was added and implemented according to RFC-1105 with minor exceptions.

■ Enhanced Exterior Gateway Protocol (EGP)

The Exterior Gateway Protocol was rewritten to be fully conformant with RFC-904. All modes are supported.

■ Enhanced Routing Information Protocol (RIP)

The Routing Information Protocol is now compliant with RFC-1058, and now correctly supports flash updates.

■ Online Configuration Changes

The user is now able to disable or enable any gated feature, including complete protocols by signal handling (without killing and restarting `gated(1M)`).

■ Increased control over ICMP redirects.

The user can now specify additional options to control ICMP redirects providing more flexible handling of these packets.

■ Improved Tracing

The tracing facility was expanded by providing tracing options with detailed output available for monitoring packets and state machines.

■ New State Table Diagnostic Tool

State dumps are now possible while `gated` is running (via `signal`).

Impact

People who use the previous version of **gated** should convert their old configuration file to the new configuration file format. A conversion tool is provided that aids in the conversion process. For more information on converting the previous configuration file for **gated** to the new format, see the **README.conv** file in the **/etc/newconfig/gated** directory.

HP ARPA Services: DTC Utilities

Dedicated Port Configuration File

The services include a file named **/etc/dp_file**, which contains Datacommunications and Terminal Controller (DTC) port information. This file is used by the system administrator to create mappings between DTC ports and pty device special file names. This information is used for two purposes:

- Login access via Telnet
- Outgoing connections to DTC ports via Telnet

The following sections describe the functionality associated with the Dedicated Port Configuration File.

Login Access Via Telnet

Telnet Option Port ID is a new feature of ARPA/9000 that facilitates login access to HP 9000 systems from HP Datacommunication and Terminal Controllers (DTCs). Telnet Option Port ID allows the system administrator to configure a correspondence between pty device special file names and DTC ports so that the source of inbound Telnet sessions from DTC ports can be determined and the connection can be assigned to a specific pty device special file. Previously, all Telnet connections were assigned pty device special files randomly from a pool.

Telnet Option Port ID manages incoming connections from the DTC to the system.

Telnet Option Port ID depends upon a configuration file, `/etc/dp_file`. This file contains mappings between DTC IP address/board/port and pty names.

Telnet Option Port ID uses the IAC DO ENVIRON Telnet command to initiate an exchange of information between the DTC and the host. When a DTC requests a Telnet session with the ARPA/9000 host, the host sends an IAC DO ENVIRON command to the DTC, which responds with IAC WILL ENVIRON followed by the board and port number of the calling port on the DTC.

The DDFA software creates information used by `telnetd` to determine if incoming Telnet connections are coming from DTC ports that have been defined in the `dp_file`. If an incoming Telnet connection is from a DTC port defined in the `dp_file`, then the well-known pty defined there is used to service the connection. If the incoming connection is not from a DTC port defined in the `dp_file`, then the pty is assigned in the traditional manner (randomly from a pool).

DTC Device File Access (DDFA) Utilities Software Component

The DTC Device File Access (DDFA) software allows access from HP-UX systems and user-written applications to HP DTCs using standard HP-UX structures. DDFA provides an interface to remote (LAN-connected) DTC ports, which is similar to the interface for local MUX ports. DDFA is a group of configuration files, executable files, and a daemon.

DDFA is documented in the **DTC Device File Access Utilities Manual** and in the following entries in the HP-UX Reference:

<code>ddfa(7)</code>	DTC Device File Access Utilities description
<code>dp(4)</code>	Dedicated port file
<code>dpp(1m)</code>	Dedicated port file parser
<code>ocd(1m)</code>	Outbound connection daemon
<code>ocdebug(1m)</code>	Debug version of ocd
<code>pcf(4)</code>	Port configuration file

DDFA Utilities are automatically installed with ARPA Services. Refer to the **Administering ARPA/9000 Services** document.

DDFA is intended for DTC users who need access to DTC devices by using pty device file names. Without DDFA, pty device file names are assigned randomly to devices on DTCs. This randomness makes it difficult to associate a specific pty device file with a specific device connected to a DTC.

DDFA allows the system or network administrator to configure a correspondence between the DTC IP address, board, and port with specified pty device file names. Then, devices on the DTC can be referred to by their pty device file names. For example, these pty device file names can be used by the application programmer with standard `read()`, `write()`, `open()`, `close()`, and `ioctl()` calls. Another example is using the HP-UX spooler to identify a DTC printer. Previously, because DTC devices were assigned random pty device file names, a DTC device could not be specified with the HP-UX spooler. With DDFA configuration, a pty device file name for a DTC printer can be specified with the HP-UX spooler.

Dedicated Port Configuration File

The correspondence, or mapping, between the DTC and pty device file names is created in a configuration file, `/etc/newconfig/ddfa/dp`, which is called the **dp file**. See the `dp(4)` entry in the HP-UX Reference. This is a master template file that the user should copy to `/etc/ddfa/dp`.

The **dp** file is part of DDFA, but it is also used by Telnet to identify incoming connections. This feature is known as **Telnet Port Identification** or **DTC Port Identification via Telnet**. An incoming connection is one that originates from the DTC.

The **dp** file is parsed by the dedicated port parser program, `/etc/dpp`, which spawns an Outbound Connection Daemon, `/etc/ocd`. The `ocd` daemon manages the connection and data transfer to each DTC port. The debug version of `/etc/ocd` is `/etc/ocdebug`.

In addition, there is a DDFA port configuration file, `/etc/newconfig/ddfa/pcf`. This is a master template file that the user should copy to `/etc/ddfa/pcf`. This file contains timer, connection, and data information for each output device specified in the **dp** file. Therefore, the **pcf** file is required only for outgoing connections (connections initiated from the host to the DTC). The `ocd` daemon uses this information to manage the Telnet connection from the pty device file to the DTC port.

4-12 Other Topics

Telnet Port Identification

Communication between the DTC and the host is done via Telnet. Previously, all Telnet connections were assigned pty device files randomly from a pool of available ptys. Enhancements to the Telnet daemon, `telnetd`, enable Telnet to use the DTC board and port and pty device file configuration information set up by DDFA in the `dp` file. The `dp` file is parsed by the `/etc/dpp` program, which is also part of DDFA. Telnet uses a binary lookup file created by `/etc/dpp`.

If an incoming Telnet connection is from a DTC port defined in the `dp` file, then the well-known pty device file defined there is used to service the connection. If the incoming connection is not from a DTC port defined in the `dp` file, then the pty is assigned in the traditional manner, randomly from a pool of available ptys.

Impact

There is no impact on ARPA/9000 users or administrators who are not specifically interested in using these features. For more information, see the DTC Device File Access Utilities document and the Administering ARPA/9000 Services document.

Computer Affected

All HP 9000 Series 300, 400, 700, and 800 computers are affected.

HP-UX FORTRAN/9000 9.0 Release (f77)

New Product Features

The new features available with the 9.0 release of the Series 800 FORTRAN 77 language are:

- Optimizing Preprocessor (`ftnopp`).
- Profile-based optimization.

- BSD compatibility library (`libU77`).
- C preprocessor support.
- `+autodblpad` option.
- Selected FORTRAN 90 Features.
- Fortran inter-procedural checker (`lintfor`).
- Changes in default compiler behavior to be consistent with Series 700 FORTRAN.

For a detail listing of the new features for the 9.0 release of the Series 800 FORTRAN 77 language, see the HP-UX FORTRAN/9000 9.0 Release Notes document and the file `Fortran` in the directory: `/etc/newconfig/90RelNotes`.

Related Documentation

The `f77(1)` command in the HP-UX Reference provides a summary reference to this command.

Other related documentation:

- HP-UX FORTRAN/9000 Programmer's Reference (B2408-90010)
- HP-UX FORTRAN/9000 Programmer's Guide (B2408-90009)
- Programming on HP-UX (B2355-90026)
- HP-UX Portability Guide (B2355-90025)

Image Viewer

The HP-UX 9.0 release includes a new version of `imageview`, the bit-mapped image file viewer. Now working on all HP 9000 Series 700 and 800 systems, `imageview` has these new features:

- Access to wide range of bit-mapped image files
- Enhanced user interface
- New hypertext help windows with sample images

- New sample image files

Wider Range of Bit-Mapped Images

While the first version of **imageview** displayed only TIFF image files, the new **imageview** displays multiple types of bit-mapped image files. The following table shows the file type, contents, and supported version:

File Type	File Contents	Version Supported
-----	-----	-----
TIFF	PC, scanned, and FAX images	5.0 (6.0 for TIFF JPEG)
JFIF	JPEG-compressed images	8-R8
GIF	xv and xgif images	87a
XWD	Pixmap images from xwd (Z format)	X11
XBM	Bitonal X bitmap images	X11
XPM	Color X pixmap images	3.0
BMF	Starbase bitmap images (Z format)	1

The following types of TIFF 5.0 images can be displayed by **imageview**:

- Bitonal (1-bit)
- Grayscale (4 and 8-bit)
- Palette (4 and 8-bit color)
- RGB (12 and 24-bit color)
- YCbCr (24-bit color)

TIFF images may be in uncompressed format or in any of the following compressed formats:

- JPEG (Grayscale, RGB, and YCbCr images)
- LZW (Bitonal, Grayscale, RGB, and YCbCr images)
- CCITT Group 3 and 4, TIFF-compatible Group 3 (Bitonal images)
- Packbits (Grayscale, RGB, Bitonal images)

Enhanced User Interface

The user interface for `imageview` was enhanced by these changes to the View menu:

- The `Fit To Window` menu item now lets you resize an image by resizing the HP VUE window.
- The `Do Not Rescale` menu item, which replaces the `Actual Size` item, lets you fix the image size separately from the window size.

Also, `imageview` now contains simple front panel controls that activate these functions:

- Show one pixel per image file pixel
- Zoom In and Zoom Out
- Change page

Full details on all menu items and controls are provided in the online help system.

New Sample Image Files

In experimenting with `imageview`, you can now display these sample TIFF image files located in the `/usr/imaging/images` directory:

Filename	TIFF Format
-----	-----
<code>example.tif</code>	Bitonal
<code>color_jpeg.tif</code>	YCbCr

The `color_jpeg.tif` file is an image converted to YCbCr and compressed by JPEG.

Starting imageview

To start `imageview` at the HP-UX command line, type:

```
/usr/bin/X11/imageview
```

For a list of `imageview` command options, see the `imageview(1)` entry in the HP-UX Reference.

Using imageview with HP VUE

The `/usr/imaging/imageview.doc` has information about the following methods of quick access to `imageview` through HP VUE 3.0:

- Double Clicking on Image Files
- Using the HP VUE 3.0 Toolbox
- Setting up an Image Drag and Drop

Other Imaging Capability

Imageview is an application developed by using the Image Developer's Kit, an HP-UX product that is available from Hewlett-Packard for use on Series 700 and 800 systems. To develop other imaging applications, ask your HP representative for information about obtaining this kit.

Instant Ignition

All new pre-installed systems are shipped configured with Logical Volumes instead of the HP-UX hard partitioning scheme. New software installations of HP-UX 9.0 default to being configured with Logical Volumes. Update customers are strongly encouraged to migrate from the HP-UX hard partitioning scheme to Logical Volumes. HP strongly recommends that a full understanding of LVM Concepts be obtained by reading the LVM chapter in the *How HP-UX Works: Concepts for the System Administrator* document before migrating to Logical Volumes.

At 9.0, the software preloading process for the Series 800, Model 8x7S computers changed. Now, only the root disk is preloaded with swap space and the file system. This process differs from the 8.02 release, in which auxiliary disks were also preloaded with swap space and a file system mount point.

The IGNITION Fileset

Boot-up Checklist

The boot-up checklist now displays **busy**, **OK**, and **FAIL** in yellow, green and red, respectively, to indicate the status of initiating various pieces of functionality.

Demos

For Series 700 systems only, the **GLANCE** and **SHARED-X** demos shipped with the Glance product on factory-ignited 8.07 systems are not supported on HP-UX 9.0 and are removed by the 9.0 IGNITION customize script.

Utilities

- The Instant Ignition utilities were moved from the **miscellaneous** directory of the application manager to the **Office_Apps** directory of the General toolbox.
- The **XHPCALC** action was renamed **Calculator**.
- The **DATEBOOK** action was renamed **Datebook**.
- The **XCAL** action was renamed **Calendar**.

For Series 700 systems only, the **XLINE** utility shipped with HP-UX 8.07 is not supported on HP-UX 9.0 and is removed by the 9.0 IGNITION customize script. Xline for 9.0 systems can be obtained by ordering the Glance product.

System Information

An action called **Peripherals** was added under the **System_Info** directory of the General toolbox. This action displays a list of the peripherals currently attached to the system.

System Administration

Actions for **Updist**, **UpdateFilesets**, **RemoveFilesets**, **RebootSystem** and **SingleUser** were added for 9.0.

The Instant Ignition system administration actions were moved from the `system_apps/sys_admin` directory of the application manager to the `System_Admin` directory of the General toolbox.

The `HALTSYS` action, which called `/usr/lib/X11/iiapps/haltsys.sh`, was replaced with the `HaltSystem` action, which calls `/usr/lib/X11/ignition/shutdown.ksh`.

Printer Configuration

When configuring a printer, SAM calls an Instant Ignition routine. This routine allows the printer to be added to the HP VUE printer subpanel.

No-windows Login

The prompts for screen-size parameters that previously occurred during login to no-windows mode were replaced by a routine that automatically sizes HP terminals.

The IGNITION-HELP fileset

Help Screens

A sequence of help screens explaining the features of an Instant Ignition system are provided with the new `IGNITION-HELP` fileset. These screens have textual and graphical hyperlinks.

Initial Login Welcome

The default VUE session was modified so that a help window is automatically opened for each user at their initial login.

Default Welcome Message

The Welcome button on the HP VUE Help subpanel is preconfigured by the `IGNITION-HELP` customize script to determine if the user is `root`. Then the user gets an appropriate welcome message.

LAN/9000

LAN Driver

The LAN Driver, **LAN3**, controls the LAN portion of the LAN/Console card as well as the LAN only card. It provides IEEE802.3/ETHERNET service and the card diagnostic functions for the Series 8x7 and 890 HP-UX system.

LAN3 was originally designed to support the LAN only card on Series 8x7 HP-UX systems for the 8.02 release. It was modified to synchronize with Terminal I/O code (Console) to support the LAN/Console card on Series 8x7 and 890 HP-UX systems for the 9.0 release. The functionality of LAN3 for the 9.0 release is the same as for the 8.02 release. Performance of LAN3 for the 9.0 release has improved.

Support Tool Manager

Features

The Support Tool Manager is a new platform for diagnostics and support tools that provides several new features: new user interfaces, novice mode of operation, and system exerciser modules. Three intuitive, easy to use interfaces are available: a graphical interface, a menu interface, and a command line interface. These interfaces are intended for novice use of the support tools.

XSTM

This user interface is Motif-based, and is designed for use with X terminals. It provides a map of the system using icons, built-in help, and the ability to invoke all available operations (verify, diagnose, or exercise) for each hardware device. Operation states are indicated by color changes to the icons. Additional **pull-down** menus are available to view log files and abort pending operations.

MSTM

The menu interface features the same map, help, and test capabilities as **xstm**. It provides the user with a set of commands for diagnosing, verifying, and exercising the various devices on the host system. Unlike **xstm** and **cstm**, the menu lets a user test specific subsystems. For example, with a single menu selection, a user can test the disk subsystem. Additional commands are available to display current system status, abort tests, and view log files. The **mstm** utility is intended for use on all Hewlett-Packard terminals and requires no special graphics capability.

CSTM

The command line interface features the same map, help, and test capabilities as **xstm**. It provides a set of commands for diagnosing, verifying, and exercising the various devices on the host system. Additional commands are available to map the system configuration, display current system status, abort tests, and view log files. The **cstm** utility is intended for use on ASCII terminals. It requires no special graphics capability and can be script driven.

Using the Tools

To use the diagnostics and support tools, the user first invokes one of the user interface modules (CSTM, XSTM, or MSTM) and enters appropriate commands to specify that a verify, diagnose, or exercise operation is to be performed on a device.

Verify operation

If available, performs a simple test of component function, providing a **pass/fail** indication of device condition; typically, this is a first level test of a device's condition.

Diagnose operation

If available, runs a diagnostic program on the device, which is designed to detect and isolate faulty hardware on that device.

Exercise operation

If available, continuously stresses the device or subsystem; this function is useful in providing very high confidence verification, and in detecting intermittent errors.

After the operation has completed, the results are displayed by the user interface and logged in the Support Tool Manager log file.

Summary of Change

The Support Tool Manager runs on Series 800 systems with the HP-UX 9.0 release. It provides verifiers, diagnostics, and exercisers for various devices on the host system. The diagnostic user interface (**sysdiag**) is supported as the expert mode user interface. The tasks performed to verify a device in the Support Tool Manager are identical to the tasks performed by the HP-UX system verifier. Selecting the diagnose operation through the Support Tool Manager performs the same set of tests as running the diagnostic program through the on-line diagnostic system using default sections and steps. The exercise operation is a new for HP-UX.

Impact

The user is not able to run exerciser or diagnose operations unless a self-maintainer/channel support license has been installed on the system. If you feel you need to access the exercise or diagnose operations and do not have a self-maintainer/channel support license, please contact you HP Account-assigned Customer Engineer.

Performance

The exerciser operations within the Support Tool Manager are designed to continuously stress a particular device or subsystem. This function is useful in providing high confidence verification and detecting intermittent errors. All of the Support Tool Manager user interfaces (XSTM, MSTM, and CSTM) let the user run multiple exercisers concurrently. Running exercisers place an additional load on system throughput. Therefore, they should not be use indiscriminately.

Obsolescence

With the introduction of the Support Tool Manager, the HP-UX system verifier is obsoleted. The verify operations within the Support Tool Manager supersedes those previously provided by the HP-UX System Verifier.

Terminal I/O

There have been several changes in the area terminal I/O for HP-UX 9.0. Following are descriptions of the changes and new features.

New Features

- SIGWINCH and the associated `ioctl` calls (TIOCSWINSZ and TIOCGWINSZ) are now supported on all HP-UX platforms. Applications supporting the SIGWINCH signal are now notified when their window is resized.

The default action for the SIGWINCH signal is to ignore the signal. Applications that do not support SIGWINCH do not see any difference from previous releases.

- The implementation of RTS/CTS hardware flow control was enhanced to allow enabling of the feature with a bit in the device file minor number. The `termio(7)` entry in the HP-UX Reference details the multiplexer models that support hardware flow control and the minor number format in the SYSTEM DEPENDENCIES section.
- STTY was modified to set and report the status of RTS/CTS hardware flow control. The new fields are `ctsxon` and `rtsxoff`. The fields are also reported in the `stty -g` option. They are located at the end of the data structure so that output is compatible with earlier versions of stty.
- The LAN/Console personality card is now supported on S8x7 HP-UX systems. The LAN/Console card contains 2 serial ports, one for the console and the second for the remote support modem. The card provides the Access Port functionality. The second serial port may also be used as a normal

HP-UX serial port. The console and second serial ports are restricted to a maximum 9600 baud operation.

Performance

The character handling code in the Line Discipline on Series 800 systems was modified to reduce the overhead on multi-byte reads and writes. The reduction in overhead is dependent on the number of bytes read or written in a single system call. Path length measurements have shown a reduction of about 70% in the TIO path for 100 byte reads and writes. Single byte reads and writes have been measured at about a 3% reduction. These measurements are only an indication of the potential improvement. As with all performance measurements, actual CPU overhead reduction is application and system dependent.

LSSERVER

LSSERVER is the run-time component of the NetLS product. LSSERVER provides an environment that allows users to install license passwords and run products that are licensed with NetLS. For information on how this version of LSSERVER interacts with LSLOCK (the NetLS application developer's package), see the section on NetLS.

The 9.0 release of LSSERVER includes changes to the `ls_targetid` utility and corrections for reported inadequacies.

Enhancements to the `ls_targetid` utility

The `ls_targetid` utility displays output in a more functional manner. The new output shows the node's target ID, now called a permanent target ID, as follows:

```
Permanent Target ID: (target id displayed here)
```

The Permanent Target ID is derived from a permanent hardware source on the system from which the `ls_targetid` program is executed. The current sources for a permanent target id are:

- SPU ID number
- LAN link level address (LLA)

If a system has an external, removable hardware device, such as an HP HIL (pod) device, the `ls_targetid` output displays the computer's permanent target ID, plus a second ID, which is called a portable target ID.

The portable ID is derived from the external device. You can use either the permanent or portable target ID for license generation, depending on your usage needs and your product's sales agreement. In most cases, the permanent target ID is the correct ID to use for generating a license. However, you can use the portable target ID to generate a password for a nodelocked license that can be transferred to a different machine by moving the external device.

The `ls_targetid` command has the following new options:

- h or -usage Gives usage information.
- q Runs in quiet mode and omits the Permanent/Portable headings.
- v Runs in verbose mode and displays descriptive information about permanent and portable target IDs, in addition to displaying the IDs.

Corrections

The product has a new version of `stcode.db`, which contains the NetLS and NCS error codes. The new `stcode.db` allows error text to be displayed when an error occurs.

The LSSERVER installation process was changed to include creation of the `/usr/netls` directory. (To be able to run, the `netlsd` daemon requires this directory.) This corrects a condition in which the `netlsd` daemon would not start on systems that did not have LSLOCK installed.

The `netlsd` daemon startup process was changed to test for the existence of an active `llbd` daemon rather than checking for the existence of the file `llbdbase.dat`. To run, the `netlsd` daemon requires the `llbd` daemon. Previously, the `netlsd` startup checked for the existence of the `llbdbase.dat` file, instead of checking for the actual `llbd` process. This caused the `netlsd` startup to fail at times when it should have succeeded.

There is new algorithm for calculating the space needed for the current license database file (`/usr/netls/cur_db`) and for storing information within the file. The algorithm reduces the size of the `cur_db` file.

The `ls_targetid` utility contains corrections for the following conditions:

- Permissions for the `/dev/ls_targetid` file have been changed to allow access by everyone.
- The `ls_targetid` utility can now derive target IDs for HP 9000 Model 832 computers.

As well, the `ls_admin` utility no longer allows expired compound passwords to be installed, and you can now add a license while a license is checked out without corrupting the current license database file (`/usr/netls/cur_db`).

NCS 1.5.1

HP-UX 9.0 includes the Network Computing Kernel (NCK) portion of the Network Computing System (NCS) 1.5.1 product. The following sections discuss changes to NCS 1.5.1 at HP-UX 9.0.

Additions to NCS 1.5.1

This section describes features of the HP-UX release of NCS 1.5.1 that were added since the NCS 1.5.1 manuals were printed.

- The status code file `stcode.db` now contains status codes for both the NCS and NetLS products.
- When you run the `what` command on NCS utilities, the output is `nck version 1.5.1 (92/07/17)`.
- The local location broker daemon, `llbd`, can now be started and stopped using the SAM utility.
- The `ppfm.h` and `idl/c/*.h` header files in NCS/NCK have been updated for better compatibility with C++. The header files now match the ones released with the 91/10/31 version of NCS/NIDL (fixes APR 7D982895/SR

5003042317; also shipped as patches PHSS-0973 (300 series), PHSS-0974 (700 series), and PHSS_0975 (800 series.))

NCS 1.5.1 Limitations

- NCS and software that depends on NCS does not work on standalone systems whose only network interface is a loopback interface (for example, 100). To operate, NCS requires a true network interface (for example, Ethernet 1an0) that provides a valid IP address.
- On any host that uses NCS, a network interface that is configured for use with IP must have its address and netmask assigned correctly. Otherwise, NCS may fail to operate. A typical symptom of an incorrect network address is that the `lb_admin` command cannot contact the local `llbd` daemon and reports communications failures. A typical symptom of an incorrect netmask is that the corresponding broadcast address is also incorrect, causing `lb_admin` to be unable to contact the Global Location Broker and to report communications failures.

Using SAM to Start llbd

You can use SAM to start, stop, and restart (stop followed immediately by start) the `llbd` daemon. Select the:

Networking/Communications ->

line from the main menu and then select:

Services: Enable/Disable

For the daemons that appear in the popped up screen, select the line that begins:

NCS LLBD

Enabling llbd

Pull down the **Actions** menu and select **Enable**. SAM performs two actions:

- Starts an `llbd` daemon.
- Ensures that `llbd` starts automatically during future system starts.

If you select **Enable** when an **llbd** is already running, SAM does not start a second **llbd**.

Disabling llbd

Pull down the **Actions** menu and select **Disable**. SAM performs two actions:

- Stops the **llbd** daemon.
- Ensures that **llbd** does not start during future system starts.

SAM does not stop the **llbd** when no **llbd** is running.

Restarting llbd

Pull down the **Actions** menu and select **Restart**. SAM stops the **llbd** that is currently running and starts a new daemon.

When you complete the changes, pull down the **View** menu and select **Exit**.

Kernel Regeneration Messages

After you manipulate **llbd**, you may see messages about the kernel needing to be regenerated. If you did nothing but enable, disable, or restart **LLBD** during a SAM session, you should cancel the kernel regeneration. Running **llbd** does not affect the kernel.

NCS 1.5.1 Enhancements and Corrections

- The **lb_admin** command **clean** correctly handles servers that register non-nil objects; in previous releases, this command incorrectly offered to delete registrations for such servers. (Fixes APR 7D988743 /SR 1653002014.)
- The global location broker daemon, **glbd**, now handles diskless clusters properly and starts up properly in heterogeneous clusters. (Fixes APRs 7D98AC33, 7D98BF2E, 7D98E828, and 7D98F5E9, corresponding to SRs 5000658369, 5000655969, 5000656009, and 5000655977. These corrections were also shipped as the following patch to HP-UX 8.0: PHSS-0415 (300 series.))
- The library, **libnck.a**, was updated so that **rpc_\$bind** verifies that the specified network family is valid. (This correction was shipped as the

following patches to HP-UX 8.05: PHSS-0769 (300 series), PHSS-0770 (800 series), PHSS-0771 (700 series.))

- The status code file, `stcode.db`, contains the `socket_$` status codes, which were inadvertently omitted from previous releases.
- At HP-UX 8.x, if a client made a remote procedure call to a server and that server in turn made a remote procedure call to another server, the initial call from the client would sometimes time-out prematurely. The NCK runtime library in HP-UX 9.0 corrects this condition. (Corrections APR 7D98F62E /SR 5003074906.)
- At HP-UX 8.x, the NCS utilities and NCS-based applications did not function correctly on hosts having kernels that contained the `atalk` AppleTalk driver. The NCK runtime library and utilities in HP-UX 9.0 corrects this condition. (Corrections APR 44032598 /SR J600629857.)

Documentation Clarifications

This section contains errata for the already printed NCS documentation.

glbd Replication

For `glbd` replication to succeed, all replicas of the Global Location Broker Daemon must propagate via the same protocol family. This means that all replica addresses must be listed under the same protocol family in all replica lists. See Chapter 4 of **Managing NCS Software** (Order Number D-11895-E) for a discussion of starting `glbd` replicas.

Using `rrpc_$` Calls

The NCK runtime library does not accept `rrpc_$` requests that specify non-nil objects. Clients that make `rrpc_$` calls should always supply `uuid_$nil` as the object UUID in the RPC handle.

NetLS

The Network License System (NetLS) product allows software vendors to create licenses for their products in order to manage, enforce, and monitor software usage and usage policies. Once a product has been licensed with NetLS, customers use the NetLS runtime environment to manage their licensed products.

NetLS consists of two components:

- LSSERVER provides the environment that allows users to install license passwords and run products that are licensed with NetLS.
- LSLOCK provides a set of tools that allow software developers to license their products and generate license passwords.

LSSERVER 2.01.08 is bundled with the 9.0 release of HP-UX. However, LSLOCK 2.01.08 is available as a separately orderable product. Applications that are licensed with LSLOCK version 2.01 or later run on HP-UX 9.0 systems. In addition, people can install LSLOCK version 2.01 or later on an HP-UX 9.0 system.

For information about the new features and changes to LSSERVER version 2.01.08 (which is bundled with this release of HP-UX 9.0), see the section on LSSERVER. Note that to take advantage of these changes to LSSERVER, applications must be licensed by the 2.01.08 version of LSLOCK.

NFS/9000

NFS/9000 is bundled with the HP-UX operating system software.

Native Language Support (NLS)

Summary of New Features

The following items highlight the features:

- Worldwide Portability Interface (WPI)
- New `locale(1M)` and `localedef(1M)`
- Removal of `getmsg(3C)` from `libc`
- Provide `getmsg.o` for backwards compatibility
- Support five new locales:
 - `arabic.iso88596`, `greek.iso88597`,
 - `hebrew.iso88598`, `turkish.iso88599`, `thai`
- Deliver 64 new `iconv` data files.
- `Iconv` `chte` tables renamed.
- Warning of interfaces no longer supported in the next major Series 800 release.
- Locales modified.
- Miscellaneous system interface changes (such as `buildlang`, `setlocale`, `getlocale`, and `nl_langinfo`)
- New filesets have been added for the X11 windowing system.

<code>JAPANESE-X11</code>	0.3 MB
<code>KOREAN-X11</code>	0.3 MB
<code>CHINESET-X11</code>	0.3 MB
<code>CHINESES-X11</code>	0.3 MB

WPI Routines

Many of the existing XPG library interfaces work with byte values, which limits their support of multi-byte codesets. X/Open could not modify all of the library interfaces to work with wide characters, because that would break backwards compatibility with previous systems.

Instead, X/Open has a parallel set of interfaces that are functionally equivalent to the current single byte interfaces, but work with wide characters. These interfaces, collectively known as the Worldwide Portability Interfaces, are recommended for use by character-based portable applications.

Character Classification

- `wctype()`: determine whether property is valid for current locale.
- `walnum()`: is character alphanumeric? alpha?
- `iswcntrl()`: is character control?
- `iswdigit()`: is character a digit?
- `iswgraph()`: is character a graphic?
- `iswlower()`: is character lower case?
- `iswprint()`: is character printable?
- `iswpunct()`: is character punctuation?
- `iswspace()`: is character whitespace?
- `iswupper()`: is character upper case?
- `iswxdigit()`: is character hex digit?
- `iswctype()`: determines whether character has given property

File Input and Output

- `fgetwc()`: get wide character from a file
- `fputwc()`: write wide character to a file
- `getwc()`: get wide character from a file
- `getwchar()`: get wide character from stdin

- `putwc()`: write character to a file
- `putwchar()`: write wide character to stdout
- `ungetwc()`: unget wide character
- `fgetw()`: get wide character string from a file
- `fputws()`: write wide character string to a file
- `getws()`: get wide character string from file
- `putws()`: write wide character string to file
- `fprintf()`, `printf()`, `sprintf()`: %C, %S conversion specifiers added for wide character or wide string formatted output
- `fscanf()`, `scanf()`, `sscanf()`: %C, %S conversion specifiers added for wide character or wide string formatted input

Case Conversion

- `towlower()`: converts wide character to lower case
- `towupper()`: converts wide character to upper case

String Manipulation

- `wscat()`: concatenate two wide character strings
- `wchr()`: find character in wide character string
- `wscmp()`: compare two wide character strings
- `wscopy()`: copy one wide character string to another
- `wscspn()`: finds section of wide character string not in set
- `wcslon()`: determine the length of wide character string
- `wcsncat()`: concatenate wide character strings up to length
- `wcsncmp()`: compare wide character strings up to length
- `wcsncpy()`: copy wide character strings up to length
- `wcspbrk()`: locate first occurrence of wide character in wide char string
- `wcsrchr()`: find character starting from end of wide character string

- `wcsspn()`: finds section of wide character string in set
- `wcstok()`: find next token in wide character string
- `wcswcs()`: find one wide character string within another
- Locale Specific String Conversions
- `wcstod()`: convert wide character string to double-precision number
- `wcstol()`: convert wide character string to long
- `wcstoul()`: convert wide character string to unsigned long

Sorting and Collation

- `wcscoll()`: wide character string collate
- `wcsxfrm()`: wide character string transform
- `wcsftime()`: convert date and time to wide character string
- `wcswidth()`: number of column positions of a wide character string
- `wcwidth()`: number of column positions of a wide character code

Impact

Applications now have a set of XPG4 portable tools to enable their application to be character set independent and operate on other XPG4 conformant systems.

Alternatives

Although the wide character interfaces adhere to XPG4 standards and are preferred for portability, multibyte interfaces in `nl_tools_16(3C)` are available only on HP-UX systems. Interfaces in `multibyte(3C)` allow for converting between multibyte and wide character.

Performance

No performance degradation of applications using the WPI routines is expected over the same applications using the existing multi-byte routines.

Size requirement

The WPI routines are available in the archived library, `/usr/lib/libwpi.a`, and the shared library, `/usr/lib/libwpi.sl`, in the NLS-CORE files, which is currently 472 KBs. Combined, the WPI libraries occupy about 120 KBs of disk space.

The memory usage of an application using WPI routines cannot be predicted, as it is entirely application dependent. However, a specific statement can be made regarding the size of a character. A normal character is one byte; whereas, a WPI character is four bytes.

New Locale(1M) and Localedef(1M)

To conform to POSIX standards, `locale(1M)` and `localedef(1M)` commands now conform to the specifications dictated by the POSIX 1003.2 standard.

The `localedef(1M)` and `locale(1M)` commands are in `/usr/bin`. The new compatible locale files are built with the `localedef(1M)` utility and named `localef` in the respective language directories of `/usr/lib/nls`.

Impact

The format of the scripts accepted by the new `localedef(1M)` and the old `buildlang(1M)` is very different. Thus, the locale input scripts that work with the old `buildlang(1M)` utility are not compatible with the new `localedef(1M)` and vice versa. To maintain backwards compatibility with existing executables, the existing `locale.def` files are also delivered to HP-UX release 9.0 in addition to the new `locale.inf` files.

Applications compiled with archived `libc` prior to HP-UX release 9.0 accessed the old `locale.def`. Those compiled with shared libraries or with HP-UX 9.0 now access the new `locale.inf` files.

Changes to the POSIX definition of `LC_CTYPE` character classification for alpha, print, and graph in `localedef` input scripts are only realized in applications that are linked with the shared library version of `libc` or recompiled with the 9.0 version of `isalpha()`, `isprint()`, and `isgraph()`.

Alternatives

Buildlang(1M) and its compatible locale files still remain on the 9.0 release for backwards compatibility.

Performance

The performance of **localedef** is not expected to degrade other than the extra time necessary to read in the few new **LC_*** data items that did not exist in the old locale input scripts. Nevertheless, **localedef(1M)** is executed only once per locale, so the performance of the **localedef(1M)** command is not critical.

Size requirement

The new locale files, **locale.inf**, need to be included on the disk along with the existing locale files, **locale.def**, for backwards compatibility with existing applications. The **locale.inf** files exist in the same directory as the existing **locale.def** files. Each existing locale file is about 3-4 KBs and each new locale file is about 4-5 KBs. Although there are currently 49 locales on HP-UX and this number doubles. The impact on the disk is minimal, since only a few locales are generally loaded on any one system.

The size of **localedef(1M)** is estimated to be 85 KBs and the size of **locale(1M)** is 54 KBs. Both commands are delivered in the NLS-CORE files set, which is currently 472 KBs.

Memory size is not an issue for the operation of **locale(1M)**, **localedef(1M)**, or the utilities that load the new locale data.

buildlang(1M)

Support for **buildlang(1M)** in this release is limited to the **-d** option, which lets users dump existing, pre-9.0, customized locales into a format that is readable and can be used as a reference to modify the new **locale.inf** locales.

The **buildlang(1M)** **locale.def** files are incompatible with the HP-UX 9.0 NLS utilities and subsystem, but are provided to allow applications that have not been recompiled or re-linked with the 9.0 **libc** to continue to function as expected.

The **buildlang(1M)** utility, its **locale.def** files, and **nlsinfo(1)** will not be supported or delivered in the next major release as **localedef(1M)**,

`locale(1)`, and its `locale.inf` files are the preferred, standard, locale tools that are compatible with the current NLS subsystem.

LC_ALL Category Items

`YESSTR` and `NOSTR`, were removed in favor of the X/Open preferred `LC_MESSAGES` items named `YESEXPR` and `NOEXPR`.

getlocale(3C)

`Getlocale(3C)` returns the `LOCALE_STATUS`, `MODIFIER_STATUS`, and `ERROR_STATUS` for the `LC_MESSAGES` category.

localeconv(3C)

To comply with standards, the following items, which were previously stored as character, are now stored as integers:

```
int_frac_digits, frac_digits, p_cs_precedes, p_sep_by_space,  
n_cs_precedes, n_sep_by_space, p_sign_posn, n_sign_posn.
```

The storage space used by these items in the `lconv` structure in `locale.h` is still character, but the data is integer.

nlsinfo(1)

This command will be obsoleted in the next release. The command that replaces it is `locale(1)`.

nl_langinfo(3C)

The following items are now recognized:

`YESEXPR` Affirmative response expression. This will replace `YESSTR` in the future.

`NOEXPR` Negative response expression. This will replace `NOSTR` in the future.

`T_FMT_AMPM` The am/pm time format string.

ERA_D_FMT The same as ERA_FMT, which will be obsoleted in the future.

setlocale(3C)

If the value of the locale parameter of **setlocale(3C)** is the empty string and the environment variable LC_ALL is set, the environment variable LC_ALL overrides values of the other LC_* and LANG environment variables.

When the environment variable LC_CTYPE is set to **japanese** or **japanese.euc** (either directly or indirectly), **setlocale(3C)** allocates additional space from the heap for extended character classification and conversion information. This increases the amount of heap space that an application uses by 5 KBs to 8 KBs. Applications that use their own memory allocation (**malloc**) package should be aware that **setlocale(3C)** calls **malloc()** for these locales.

Examples:

An example showing the precedence of the LC_ALL environment variable :

```
setlocale(LC_ALL, "");
```

with the following settings in the users environment :

```
LANG=german
LC_ALL=french
```

All categories are loaded with **french**. Another example showing the precedence of the LC_ALL environment variable :

```
setlocale(LC_CTYPE, "");
```

with the following settings in the users environment :

```
LANG=turkish
LC_ALL=danish
LC_CTYPE=russian
```

The LC_CTYPE category are loaded with **danish**.

Another example with the LC_ALL environment variable :

```
setlocale(LC_TIME, polish);
```

with the following settings in the users environment :

```
LANG=italian
LC_ALL=dutch
```

The LC_TIME category is set to polish.

Removal of getmsg(3C)

The obsolete NLS call, `getmsg(3C)`, was removed from `libc`.

Background

The HP NLS `getmsg(3C)` programmatic call in `libc` conflicts with the AT&T Streams programmatic call, `getmsg(2)` in `/usr/lib/libstr.a`, which was added to the 9.0 release. The name of the Streams `getmsg(2)` call cannot be changed because it is an industry-wide standard. However, the impending obsolescence of the HP NLS `getmsg(3C)` call was advertised in the `getmsg(3C)` entry in the HP-UX Reference for the past 2 releases. Since an equivalent (but not portable) call can be made to `catgetmsg(3C)` or to the portable and recommended `catgets(3C)`, HP proceeded with the obsolescence of the NLS `getmsg(3C)` call.

`Getmsg(3C)` exists in `/lib/libc.a`, `/lib/libc.sl`, and `/lib/libp/libc.a`, but `libc` is used to generically refer to all three libraries rather than to mention all three `libc` files each time.

Change

`Getmsg(3C)` is removed from `libc` and `/usr/old/getmsg.o` is provided for applications requiring backwards compatibility. Applications can then link to the `/usr/old/getmsg.o` if the `getmsg(3C)` calls have not been converted into the portable `catgets(3C)` call. Applications that use the streams library should not use `/usr/old/getmsg.o`.

Impact

All applications that were linked with the `getmsg(3C)` in the shared library, `/lib/libc.sl`, in the 8.0 HP-UX release and did not modify the source code to use `catgets(3C)` needs to re-link to the `/lib/getmsg.o` file to resolve the link to the `getmsg(3C)` shared library call.

All HP-UX libraries that call `getmsg(3C)` should replace the call with calls to `catgets(3C)` or `catgetmsg(3C)`.

Applications compiled with the shared library version of NLS `getmsg(3C)` are not binary compatible with the HP-UX 9.0 system. This applies to 8.0 based releases (8._/IF_/MCF) of the S700 and the Series 800 systems.

Alternatives

Usage of the standards conformant `catgets(3C)` call is recommended as a replacement to `getmsg(3C)`.

Size Requirement

No impact foreseen.

New Locales

Six new locales are provided. Thai is treated as a new language.

ID	name	Locale name	Lang ID	Path	Name

26	ISO 8859/6	arabic.iso88596	301	/usr/lib/arabic/iso88596/	locale.inf
27	ISO 8859/7	greek.iso88597	321	/usr/lib/greek/iso88597/	locale.inf
28	ISO 8859/8	hebrew.iso88598	341	/usr/lib/hebrew/iso88598/	locale.inf
29	ISO 8859/9	turkish.iso88599	361	/usr/lib/turkish/iso88599/	locale.inf
7	THAI8	thai	91	/usr/lib/thai/	locale.inf

Impact

No impact to the application developer.

Alternatives

None.

Performance data

N/A

Size requirement

Disk space:

- 3 KBs for each optional locale loaded on a system
- 3 KBs for the optional THAI fileset overhead

Locales Modified: katakana, japanese, japanese.euc

In HP's goal to adhere to standards, such as Draft Japanese National Profile for POSIX Version 2.0 developed by IPSJ/ITSCJ SSI/POSIX WG and OSF's ja_JP locale, characters in the range 0xA1-0xDF (8-bit Katakana characters) have been deleted from the upper, lower, punct, alpha, and alnum class in LC_CTYPE and placed in the print and graph class in LC_CTYPE.

If your application does not work properly for 8-bit Katakana characters, you may want to customize the locales for katakana, japanese, japanese.euc. To do this:

1. Dump data in the locales with `localedef -d`
2. Edit the locale file created and add the characters 0xA6-0xDF in the upper, lower, alpha, and alnum classes, and add 0xA1-0xA5 in the punct class in the LC_CTYPE category.
3. Use `localedef` to install the modified locale file. See `localedef(1M)` for more information.

New iconv Data Files

To enable interoperability of data with different character encoding schemes, 64 new `iconv` files are provided.

iso8858/1 <=> ebcdic conversion (14 west European languages, 28 files)

<code>/usr/lib/iconv/direct/iso81}amere</code>	<code>/usr/lib/iconv/direct/amere}iso81</code>
<code>/usr/lib/iconv/direct/iso81}c-fre</code>	<code>/usr/lib/iconv/direct/c-fre}iso81</code>
<code>/usr/lib/iconv/direct/iso81}danie</code>	<code>/usr/lib/iconv/direct/danie}iso81</code>
<code>/usr/lib/iconv/direct/iso81}dutce</code>	<code>/usr/lib/iconv/direct/dutce}iso81</code>
<code>/usr/lib/iconv/direct/iso81}engle</code>	<code>/usr/lib/iconv/direct/engle}iso81</code>
<code>/usr/lib/iconv/direct/iso81}finne</code>	<code>/usr/lib/iconv/direct/finne}iso81</code>
<code>/usr/lib/iconv/direct/iso81}frene</code>	<code>/usr/lib/iconv/direct/frene}iso81</code>
<code>/usr/lib/iconv/direct/iso81}germe</code>	<code>/usr/lib/iconv/direct/germe}iso81</code>
<code>/usr/lib/iconv/direct/iso81}icele</code>	<code>/usr/lib/iconv/direct/icele}iso81</code>

/usr/lib/iconv/direct/iso81}itale	/usr/lib/iconv/direct/itale}iso81
/usr/lib/iconv/direct/iso81}norwe	/usr/lib/iconv/direct/norwe}iso81
/usr/lib/iconv/direct/iso81}porte	/usr/lib/iconv/direct/porte}iso81
/usr/lib/iconv/direct/iso81}spane	/usr/lib/iconv/direct/spane}iso81
/usr/lib/iconv/direct/iso81}swede	/usr/lib/iconv/direct/swede}iso81

iso8859/2 <=> ebcdic conversion (6 east European languages, 12 files)

/usr/lib/iconv/direct/iso82}czece	/usr/lib/iconv/direct/czece}iso82
/usr/lib/iconv/direct/iso82}hunge	/usr/lib/iconv/direct/hunge}iso82
/usr/lib/iconv/direct/iso82}polie	/usr/lib/iconv/direct/polie}iso82
/usr/lib/iconv/direct/iso82}rumae	/usr/lib/iconv/direct/rumae}iso82
/usr/lib/iconv/direct/iso82}serbe	/usr/lib/iconv/direct/serbe}iso82
/usr/lib/iconv/direct/iso82}slove	/usr/lib/iconv/direct/slove}iso82

iso8859/5 <=> ebcdic conversion (2 Cyrillic languages, 4 files)

/usr/lib/iconv/direct/iso85}russe	/usr/lib/iconv/direct/russe}iso86
/usr/lib/iconv/direct/iso85}bulge	/usr/lib/iconv/direct/bulge}iso86

iso8859/6 <=> ebcdic conversion (Arabic, 2 files)

/usr/lib/iconv/direct/iso86}arabe	/usr/lib/iconv/direct/arabe}iso86
-----------------------------------	-----------------------------------

iso8859/7 <=> ebcdic conversion (Greek, 2 files)

/usr/lib/iconv/direct/iso87}greee	/usr/lib/iconv/direct/greee}iso87
-----------------------------------	-----------------------------------

iso8859/8 <=> ebcdic conversion (Hebrew, 2 files)

/usr/lib/iconv/direct/iso88}hebre	/usr/lib/iconv/direct/hebre}iso88
-----------------------------------	-----------------------------------

iso8859/9 <=> ebcdic conversion (Turkish, 2 files)

/usr/lib/iconv/direct/iso89}turke	/usr/lib/iconv/direct/turke}iso89
-----------------------------------	-----------------------------------

arabic8 <=> iso8859/6 conversion (Arabic, 2 files)

/usr/lib/iconv/direct/arab8}iso86	/usr/lib/iconv/direct/iso86}arab8
-----------------------------------	-----------------------------------

greek8 <=> iso8859/7 conversion (Greek, 2 files)

/usr/lib/iconv/direct/gree8}iso87	/usr/lib/iconv/direct/iso87}gree8
-----------------------------------	-----------------------------------

hebrew8 <=> iso8859/8 conversion (Hebrew, 2 files)

`/usr/lib/iconv/direct/hebr8}iso88` `/usr/lib/iconv/direct/iso88}hebr8`

turkish8 <=> iso8859/9 conversion (Turkish, 2 files)

`/usr/lib/iconv/direct/turk8}iso89` `/usr/lib/iconv/direct/iso89}turk8`

thai8 <=> ebcdic conversion (Thai, 2 files)

`/usr/lib/iconv/direct/thai8}thaie` `/usr/lib/iconv/direct/thaie}thai8`

big5 <=> roc15 conversion (Chinese, 2 files)

`/usr/lib/iconv/direct/big5}roc15` `/usr/lib/iconv/direct/roc15}big5`

Impact

No impact to the application developer.

Alternatives

None.

Performance data

N/A

Size requirement

An additional 20 KBs for the 64 `iconv` tables in the optional fileset NLS-CORE (currently 472 KBs)

Iconv tables Modified (Roman8 to EBCDIC IBM3172 Conversion)

The `iconv` translation tables for Roman8<—>EBCDIC were modified to apply the same conversion rule as Roman8<—>ISO 8859/1, especially for those characters not mapped 1-to-1. This makes the conversion consistent between Roman8<—>EBCDIC and Roman8<—>ISO<—>EBCDIC.

The following files in the directory, `/usr/lib/iconv/direct`, were modified:

<code>amere}roma8</code>	<code>c-fre}roma8</code>	<code>danie}roma8</code>	<code>dutce}roma8</code>
<code>engle}roma8</code>	<code>finne}roma8</code>	<code>frene}roma8</code>	<code>germe}roma8</code>
<code>icele}roma8</code>	<code>itale}roma8</code>	<code>norwe}roma8</code>	<code>porte}roma8</code>

roma8}amere	roma8}c-fre	roma8}danie	roma8}dutce
roma8}engle	roma8}finne	roma8}frene	roma8}germe
roma8}iclele	roma8}itale	roma8}norwe	roma8}porte
roma8}spane	roma8}swede	spane}roma8	swede}roma8

Roman8 to EBCDIC Conversion

Hex Value	Character	EBCDIC Character
0xA9	Accent grave	Left single quote
0xAA	Circumflex accent	caret
0xAC	Tilde accent	Tilde
0xAF	Italian Lira	Pound sign
0xB0	Overline	Macron
0xBE	Dutch guilder	Lowercase f
0xEB	Uppercase S caron	Uppercase S
0xEC	Lowercase s caron	Lowercase s
0xEE	Uppercase Y umlaut	Uppercase Y
0xF6	Long dash	Hyphen
0xFC	Solid Box	Asterisk

Roman8 to EBCDIC per Language

Roman8	Hex	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	Am	CF	Da	Du	En	Fi	Fr	Gr	It	No	Po	Sp	Sw	Ic	
A9	79	79	79	79	79	51	A0	79	DD	79	79	79	51	8C	
AA	B0	B0	5F	B0	BA	5F	5F	5F	5F	5F	B0	BA	5F	EC	
AC	A1	A1	DC	A1	BC	DC	BD	59	58	DC	A1	BD	DC	CC	
BB	B1	B1	B1	B1	5B	B1	7B	B1	7B	B1	B1	B1	B1	B1	
BE	86	86	86	86	86	86	86	86	86	86	86	86	86	86	
EB	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	
EC	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	
EE	E8	E8	E8	E8	E8	E8	E8	E8	E8	E8	E8	E8	E8	E8	
F6	60	60	60	60	60	60	60	60	60	60	60	60	60	60	
FC	5C	5C	5C	5C	5C	5C	5C	5C	5C	5C	5C	5C	5C	5C	

Key:

Am 1=american	CF 2=c-french	Da 3=danish
Du 4=dutch	En 5=english	Fi 6=finnish
Fr 7=french	Gr 8=german	It 9=italian
No 10=norwegian	Po 11=portuguese	Sp 12=spanish
Sw 13=swedish	Ic 14=icelandic	

EBCDIC to Roman8 Conversion

EBCDIC Character	Hex Value	Character
Cedilla	0x2C	Comma
Soft hyphen	0x2D	Hyphen
Division sign	0x2F	Solidus
Superscript one	0x31	Digit one
Superscript two	0x32	Digit two
Superscript three	0x33	Digit three
Registered Trade Mark	0x52	Uppercase R
Copyright sign	0x63	Lowercase c
Multiplication sign	0x78	Lowercase x
Broken bar	0x7C	Vertical line
Not sign	0x7E	Tilde

Alternatives

None.

Performance data

N/A

Size requirement

None.

EBCDIC to Roman8 code conversion for each language

															Roman8
1	2	3	4	5	6	7	8	9	10	11	12	13	14	Hex	
Am	CF	Da	Du	En	Fi	Fr	Gr	It	No	Po	Sp	Sw	Ic	Value	
9D	9D	9D	9D	9D	9D	9D	9D	9D	9D	9D	9D	9D	9D	2C	
CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	2D	
E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	2F	
DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	31	
EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	32	
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	FA	33	
AF	AF	AF	AF	AF	AF	AF	AF	AF	AF	AF	AF	AF	AF	52	
B4	B4	B4	B4	B4	B4	B4	B4	B4	B4	B4	B4	B4	B4	63	
BF	BF	BF	BF	BF	BF	BF	BF	BF	BF	BF	BF	BF	BF	78	
6A	6A	70	6A	6A	CC	DD	CC	CD	70	6A	49	CC	6A	7C	

```
|5F| 5F| BA| 5F| 5F| BA| BA| BA| BA| BA| 5F| 5F| BA| BA|| 7E |
|-----|-----|
```

Key:

```
Am 1=american      CF 2=c-french      Da 3=danish
Du 4=dutch          En 5=english      Fi 6=finnish
Fr 7=french         Gr 8=german        It 9=italian
No 10=norwegian     Po 11=portuguese     Sp 12=spanish
Sw 13=swedish       Ic 14=icelandic
```

Iconv chte tables renamed

The iconv chte tables were incorrectly named and are being renamed to allow them to work with the iconv utilities and interfaces. The following files are affected:

Old Filename	New Filename
-----	-----
/usr/lib/iconv/direct/chte}roc15	/usr/lib/iconv/direct/chine}roc15
/usr/lib/iconv/direct/roc15}chte	/usr/lib/iconv/direct/roc15}chine

HP-UX Proprietary Interfaces to be Obsoleted in Next Release

The following HP NLS routines will be obsolete in the next major Series 800 HP-UX release. These routines will still be provided in their existing locations for backwards compatibility in HP-UX, but support, enhancements, and defect fixes will terminate at that release. The use of the standards conformant routines under the column labeled, Recommended Routine(s), is highly recommended.

All of the obsolete routines identified do not follow current standards as prescribed/adopted by X/Open, Portable Operating System Interface (POSIX), System V Interface Document (SVID), and Open Software Foundation (OSF)

Obsolete Routines

```
-----
```

Obsolete Recommended

```
Routine Routine Notes
```

```
-----
```

```
builddlang localedef
catgetmsg catgets
catread catgets
currlangid no longer needed
fprintmsg fprintf
```

getmsg catgets conflict with Streams getmsg
 ICONV iconv XPG4 interface coming
 iconvclose iconv_close XPG4 interface coming
 iconvopen iconv_open XPG4 interface coming
 iconvsize
 idtolang no longer needed
 langid no longer needed
 langinfo nl_langinfo
 langinit setlocale
 langtoid no longer needed
 msgbuf.h
 n-computer C or POSIX language name
 nl_asctime strftime
 nl_asctime strftime
 nl_atof atof
 nl_catopen catopen
 nl_ctime strftime
 nl_cxtime strftime
 nl_fprintf fprintf
 nl_fscanf fscanf
 nl_gcvrt gcvrt
 nl_init setlocale
 nl_isalnum isalnum
 nl_isalpha isalpha
 nl_iscntrl iscntrl
 nl_isdigit isdigit
 nl_isgraph isgraph
 nl_islower islower
 nl_isprint isprint
 nl_isspace isspace
 nl_isupper isupper
 nl_isxdigit isxdigit
 nl_msg catgets
 nl_printf printf
 nl_scanf scanf
 nl_sscanf sscanf
 nl_sprintf sprintf
 nl_strcmp strcoll
 nl_strncmp strcoll
 nl_strtod strtod
 nl_tolower tolower
 nl_toupper toupper
 nlsinfo locale
 printmsg printf
 sprintmsg sprintf
 strcmp8 strcoll
 strcmp16 strcoll
 strncmp8 strcoll

strncmp16 strcoll

HP-UX Proprietary Commands and Interfaces

The following table describes some NLS commands and library calls that exist only on HP-UX systems. Use of these utilities and routines in applications may not be portable to other vendors' platforms and may be obsoleted in the future as standard interfaces and commands emerge. WPI should be used instead of many of the library calls listed below.

Commands:

dumpmsg
findmsg
findstr
forder
insertmsg
nljust

Library calls:

ADVANCE
advance
BYTE_STATUS
byte_status
C_COLWIDTH
c_colwidth
CHARADV
CHARAT
FIRSTOF2
firstof2
getlocale
ICONV1
ICONV2
iconvlock
PCHAR
PCHARADV
SECOF2
secof2
strord
WCHAR
WCHARADV

PortNLS

The PortNLS product (`/usr/lib/libportnls.a`), which was originally developed as a temporary solution for porting between MPE and HP-UX, does not adhere to existing standards and will be obsoleted in future releases. Since both MPE and HP-UX are striving towards providing standards based interfaces, the need for this library has been obsoleted.

Pascal Language

With the 9.0 release, many languages are available as separate products. As such, the products have their own release notes. The release notes for the products are placed on the system during the update process. They appear, after the update, in the following directory:

`/etc/newconfig/90RelNotes`

This situation applies to the Pascal language.

Starbase (Series 800 Only)

Starbase for multi-user Series 800 HP-UX 9.0 is only available to customers with update support services (BASIC). The following graphics device types are supported on the Series 800 through update support services:

- HP-CGM
- HP-GL
- HP-GL CADplt
- HP-GL CADplt2
- Keyboards
- Memory
- Terminals
- SOX 11

Bit-mapped display drivers, including the SRX (HP 98720-21), TurboSRX (HP 98730-31), and VRX color (HP 98550-56), are NOT supported on the Series 800 multi-user systems. For more information, see the Kernel and Series 800 Graphics Support topic.

The recommended graphics substitute for the Series 800 multi-user systems is Xlib. For more information, see the HP X Windows documentation, especially **Programming with Xlib**.

Other ways to perform graphics on the Series 800 multi-user systems are by attaching an X terminal and using SOX11 to display graphics; or by attaching plotters to the HP-IB for drawings.

Starbase and the 9.0 Math Libraries (Series 300/400/700/800)

Starbase applications compiled or recompiled on HP-UX 9.0 or later **MUST** link the math library, `-lm`, at compile time. This is true for both shared and archive libraries.

Starbase applications compiled on HP-UX versions 8.07 or previous versions do not need to link in the math library unless the application specifically uses it.

The 9.0 version of the Starbase Device Drivers Manual, P/N B2355-90019, now provides an example of how to link a Starbase graphics program to display output in an X11 window for each device driver. Each example lists the `-lm` except for CADplt. The `-lm` math library should be added to the CADplt linking example as well.

Switchover/UX

Features

SwitchOver/UX was introduced in release 8.0. It provides near-continuous operation of the system by detecting system failures and allowing for a switchover between a primary and a standby processor.

In release 9.0, SwitchOver/UX was enhanced to support various new interfaces, disks and technologies.

Summary of Change

SwitchOver does support two new disk interfaces: HP-PB Fiber Link and HP-PB SCSI. SwitchOver does support HP-PB FL on all HP-PB configurations. New root disks supported in HP-PB FL configurations include FL disk arrays (HP C2250) in all of their various configurations.

SwitchOver does support HP-PB SCSI in limited configurations. SCSI is only supported on the 8x7 and 890 models. Symmetric configurations are limited to one primary and one standby system. Asymmetric configurations can have up to four primaries backed up by one standby system. There are limitations to the types of disks supported in SwitchOver SCSI configurations.

Mixing disk interface types within SwitchOver configurations is not supported.

SwitchOver supports the new FDDI network interface for host to host communication. One or two FDDI interfaces may be used. The second interface allows for redundant network communication between hosts. A mix of FDDI and LAN cards is not supported.

SwitchOver supports use of the Logical Volume Manager. This includes support of root logical volumes, and LVM disk mirroring.

Impact

The only user-level change to the SwitchOver product is the addition of three new optional variables in the Switchinfo file to be used in the case of a root logical volume. Information regarding the root device and it's mirror is kept in the SwitchOver configuration file, **Switchinfo**. With LVM, an additional mirror of the root device is allowed. Information regarding this additional boot device is kept in the new variables **rootthird**, **rootthird#**, and **bootthird**.

Performance

There should be no performance impact as a result of these enhancements.

HP-UX Symbolic Debugger 9.0 Release (xdb)

New Product Features

The new features available with the 9.0 release of the HP-UX symbolic debugger are:

- Additional support for debugging shared libraries, including symbolic debugging.
- Support for C++ 3.0.
- Support for screen mode on additional terminal types and X Window resizing.
- Enhanced stack/local variable viewing.
- Enhanced source file location handling.

For a detail listing of the new features for the 9.0 release of the HP-UX symbolic debugger, see the HP-UX Symbolic Debugger 9.0 Release Notes document shipped with your particular language and the file **Debugger** in the `/etc/newconfig/90RelNotes` directory.

Related Documentation

The `xdb(1)` and `cdb(1)` commands in the HP-UX Reference provide information about these commands.

The HP-UX Symbolic Debugger User's Guide (B2355-90044) describes the features of the symbolic debugger.

X11 Windows Dev. Toolkit

See X Graphical topic.

X11 Window System

See X Graphical topic.



Copyright © 1992
Hewlett-Packard Company
Printed in USA E0892

**Manufacturing
Part No.
B3108-90002**



B3108-90002