

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

HP 9000 Computers



**HP Part No. B3782-90073
Printed in USA November 1995**

**First Edition
E1195**

Legal Notices

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

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

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

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

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

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

Copyright Notices. ©copyright 1983-95 Hewlett-Packard Company, all rights reserved.

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

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

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

©copyright 1980, 1984, 1986 Novell, Inc.

©copyright 1986-1992 Sun Microsystems, Inc.

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

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

©copyright 1986 Digital Equipment Corporation.

©copyright 1990 Motorola, Inc.

©copyright 1990, 1991, 1992 Cornell University

©copyright 1989-1991 The University of Maryland

©copyright 1988 Carnegie Mellon University

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

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

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

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

Printing History

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

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

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

Contents

1. Introduction	
How To Use this Manual	1-2
Who Should Install HP-UX 10.x?	1-3
Who Should Upgrade to HP-UX 10.x?	1-3
What If You Are on an Earlier Release?	1-3
What If You Are Already on 10.0 or 10.01?	1-3
Who Should Not Upgrade to HP-UX 10.x?	1-5
Hardware Constraints	1-5
Software Constraints	1-5
9.x to 10.10: a Two-Step Process	1-6
Upgrade to 10.01 Required	1-6
New Upgrade Capabilities	1-7
Where To Go Next	1-9
2. Planning a Site Upgrade	
Setting a Strategy	2-2
Factors To Take into Account	2-2
Need	2-2
Ease	2-2
Affinity	2-3
Interoperability	2-3
Shared HP-UX “System” Directories	2-3
A Sample Strategy	2-4
Strategy for Similarly Configured Systems	2-6
Sharing Disk Space	2-8
Operating 9.x and 10.x Systems Together	2-9
10.01 User Login Scripts	2-9
Converting HP-UX (“DUX”) Clusters	2-10
Upgrading a “DUX” Cluster	2-11
What Does the Conversion Do for You?	2-12

File Systems	2-14
Swap	2-15
Applications	2-18
If Clients Are Not Booted	2-18
Clients with “Dual Personalities”	2-20
Managing Root’s Home Directory	2-20
Providing for Series 300 and 400 Clients	2-21
Summary of “DUX” Issues	2-23
3. Planning a System Upgrade	
Who Should Not Upgrade	3-2
Systems that Cannot Be Upgraded	3-2
Upgrade Paths	3-3
Upgrade Tools	3-3
Upgrade Options	3-4
Option A: Quick Start	3-4
Time Needed for Option A	3-6
Option B: Full Preparation	3-6
Time Needed for Option B	3-8
Will You Need To “Re-Install?”	3-9
4. Pre-Upgrade Tasks for All 9.x Systems	
Summary of Upgrade Tasks From Here On	4-2
Time Frame	4-3
Timetable for Pre-Upgrade Tasks	4-4
Locating and Loading Tools and Documentation	4-5
Understanding the Changes	4-8
Before You Run Pre-Upgrade Tools	4-8
Before You Upgrade to 10.01	4-9
Changing Existing Procedures	4-10
Backups	4-10
Recommendation for Routine Backups on Existing 9.x Systems	4-11
Before You Go On	4-13
Preparing for Changes to HP-UX Clusters	4-14
What the Tools Do	4-15
What You Need to Do (Summary)	4-16
Preparing To Upgrade a 9.07 System	4-18

Analyzing Your System	4-19
What snoop Does	4-19
What You Will See	4-20
What Is Match-What-Target-Has?	4-20
What Are Bundles and Products?	4-22
When To Use Bundles	4-24
When To Use Products	4-24
Adding Software after Match-What-Target-Has	4-25
How To Add and Preview Software	4-25
Summary	4-26
Running snoop	4-27
Procedure	4-28
Modifying Your System	4-37
Modifying HP-UX Clusters	4-38
Removing Series 300/400 Clients and Inactive Clients	4-38
What if You Do Not Clean Up	4-40
Handling Non-“System” CDFs	4-41
Saving Non-“System” CDFs	4-42
Reclaiming Device Swap	4-43
Creating the Swap Reclamation File	4-44
“Unsecuring” a Trusted System	4-45
Handling snoop PROBLEMS	4-46
If You Have Less than 16 Mb of Memory	4-46
If You Have an 8Mb Memory Card in Slot One (Series 800)	4-47
If There Are “PROBLEMS” on an HP-UX Cluster Server	4-48
If You Do Not Have Enough Disk Space	4-49
Removing Unneeded Filesets	4-51
Series 800 Small System	4-51
Non-Graphics Bundles	4-52
Filesets To Remove	4-53
What To Do Next	4-56
Preparing for I/O System Changes	4-57
Before You Start	4-57
Summary of Tasks	4-58
Disks	4-59
If You Are Using Unsupported Disks	4-60
If You Are Running DataPair/800 (Series 800)	4-60

If You Want to Convert Hard-Partitioned Disks to LVM (Series 800)	4-61
1. To convert hard-partitioned non-root disks	4-61
2. To move the root filesystem from a hard partition to a new disk to be managed via LVM	4-61
3. To convert the current root disk to LVM	4-62
Other Considerations for Series 800s	4-64
Considerations for Series 700s	4-65
Device File Convention Supported for SDS conversion	4-68
Drivers	4-70
What You Need to Do	4-71
Autochanger Driver Changers	4-72
Device Files	4-75
Preparing for File System Layout Changes	4-76
What Will Happen to Non-System Files?	4-76
Preparing for Changes to Commands and Libraries	4-78
Preparing for Other Changes	4-79
Run Levels	4-79
Root User's Home Directory	4-79
Certifying Applications	4-80
In-House Applications and Scripts	4-80
HP Applications	4-80
Building an SD Depot	4-82
Setting Up a Depot on a 9.x System	4-82
If the 10.01 Software Is on Tape	4-84
More examples	4-84
Changes for Upgrades via CDROM	4-86
Multi-User Licences	4-86
LVM Mirroring	4-86
Upgrading Networking Products	4-87
Getting Ready to Upgrade	4-89
Backing Up Your System	4-89
Data Disks	4-89
Root Disk	4-90
What To Do Next	4-92

5. Converting Code and Scripts	
What If You Don't Run the Tools?	5-2
What the Tools Do	5-3
What the Tools Don't Do	5-4
Analyzing and Converting Your Files	5-5
Analyzing Files	5-5
Strategies for Running prepare and analyzer	5-5
Case 1	5-6
Case 2	5-6
Case 3	5-6
Two-Stage Cookbook for Running Prepare	5-7
Before You Start	5-7
Procedure	5-7
One-Stage Cookbook for Running Prepare	5-10
Before You Start	5-10
Procedure	5-10
Examples	5-13
Useful prepare Options: Quick Reference	5-14
Cookbook for Running Analyzer (without prepare)	5-15
Converting Files	5-16
Example	5-17
Looking up Pathnames	5-19
6. Upgrading Your System from 9.x to 10.01	
What Happens During the Upgrade	6-2
What upgrade Does for You	6-2
What upgrade Will Not Do for You	6-2
When Do You Have To Do What?	6-3
Checklist of Pre-Upgrade Tasks	6-4
Obtaining and Entering Codewords	6-5
Running the Upgrade	6-6
Pre-Flight Checks	6-6
Upgrade Procedure	6-7
Troubleshooting <code>reclaim_swap</code>	6-19
Error Messages and What They Mean	6-19
What To Do Next	6-22
Upgrading HP-UX ("DUX") Cluster Clients	6-23
Before You Start	6-23

Procedure for Upgrading Clients	6-24
What To Do Next	6-26
Running an Unattended Upgrade	6-28

7. After the Upgrade

Recovery from a Crash or Catastrophic Failure	7-2
Important: Order of Major Tasks	7-3
Tasks You Should Do Now	7-4
Tasks You Can Do Later	7-5
Setting and Using Pathnames	7-6
Setting MANPATH	7-6
Using New 10.x Pathnames	7-6
Using 9.x Pathnames in Single-User Mode	7-6
Checking Run-Levels	7-7
Examining the Upgrade Log Files	7-7
LVM Conversion “Error” in a Cluster	7-8
Making a Recovery System	7-9
Loading New 10.01 Features and HP Applications	7-10
Loading New 10.01 Features	7-10
Procedure	7-11
Loading Applications	7-13
Updating to 10.10	7-14
Updating SD	7-14
What You Are Going To Do	7-14
Using swgettools to Update SD	7-15
Disk Space	7-15
Loading swgettools	7-15
Getting the New SD Tools	7-15
Examples	7-15
Updating SD without Root Access	7-16
Running the Update	7-17
Loading 10.10 Applications	7-19
Resecuring a Trusted System Cluster	7-20
Reorganizing Your Directories	7-21
Adding Drivers	7-21
Checking Device Files	7-22
Modifying System Startup and Shutdown Scripts	7-23
Modifying /etc/inittab	7-24

Modifying User Scripts	7-25
HP VUE Configuration files	7-25
X Window System Configuration Files	7-27
HP MPower	7-27
Troubleshooting	7-28
(1)PEX	7-28
(2)After Removing Softbench Products	7-28
Checking for Known Problems	7-29
Moving Code and Scripts into Place	7-29
Testing and Certifying Applications	7-29
Converting Disks to LVM (Series 700)	7-30
Should You Convert to LVM?	7-30
How Many Disks Do You Have on this System?	7-30
Do You Need the Flexibility of LVM?	7-30
What Is Involved in Administering LVM?	7-30
Summary	7-31
Converting the Root Disk to LVM	7-31
Removing LVM	7-32
Extending File Systems	7-33
Extending Logical Volumes	7-33
Extending a Journalled File System	7-33
Removing Software You Do Not Need	7-34
Removing 10.x Software	7-34
Removing 9.x Software	7-35
Setting Up a New NFS Diskless Cluster	7-36
Maintaining Transition Links	7-36
Converting Code and Scripts	7-37
8. Compatibility between 9.x Releases and 10.01	
Applications	8-2
Guidelines for Writing Portable Applications	8-4
Specific Guidelines for Moving Code from 9.x to 10.x	8-5
Pattern-Matching Routines Mixed in Unusual Ways	8-5
Combinations with 10.x glob(3C) or 10.x regcomp(3C)	8-5
10.x wordexp(3C) and pre-10.0 wordfree(3C)	8-7
Shadow Password Routines	8-7
Results of calling password routines from 9.x executable on 10.x	8-8

HP-UX (“DUX”) Clusters No Longer Supported	8-11
NLS Routines No Longer Supported	8-13
8.x Applications Linked Archived	8-14
Other NLS Changes	8-14
NetIPC Routines No Longer Supported	8-14
Changes in Error Handling for Math Library Functions	8-17
Summary of Changes	8-17
Floating-Point to Integer Conversion	8-25
Other Changes Affecting Applications	8-26
Standards Conformance Changes	8-26
I/O Convergence Changes	8-26
Linker Defects	8-26
Procedure	8-28
Technical Details	8-29
Moving Files	8-30
Recovering, Moving or Mounting Files from a 9.x System	8-30
A. Contents of the HP-UX 10.1 “Core Media”	
Contents of 10.01	A-1
B. NFS Diskless Questions and Answers	
Cluster Configuration	B-2
Performance	B-8
Single Point Administration	B-10
Policies	B-10
Cluster-wide Tasks	B-12
File Systems	B-14
Users and Groups	B-15
Modifying the Password File Programmatically	B-16

Tables

1-1.	1-9
4-1.	4-4
4-2. Fileset Information	4-5
4-3. Online Documents Information	4-6
A-1. HP-UX 10.0 Bundles/Products	A-2

Introduction

This manual tells you how to upgrade from an HP-UX 9.x release to 10.01 or 10.10.

You need a different manual, *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*, if you want to:

- Install 10.01 or 10.10 (as opposed to upgrading); *or*
- Update from 10.0 to 10.01 or 10.10; *or*
- Update from 10.01 to 10.10.

Note

- The formal name for HP-UX release 10.0 is HP-UX 10.0 version B.10.00. The formal name for HP-UX release 10.01 is HP-UX 10.0 version B.10.01. From now on, this manual will refer to these releases as “10.0” and “10.01” respectively.
 - Throughout this manual we’ll use the term **upgrade** to describe the transition from 9.x to 10.01, and **update** to describe the smaller step from 10.0 or 10.01 to a later 10.x release.
-

How To Use this Manual

Unlike most manuals, this book is meant to be read from start to finish.

This does not mean you have to read every word and do every task, but it does mean you should go through each chapter of the book in order, understand each topic and decide whether that topic requires action on your part for the system or systems you need to upgrade.

If you do this, and you do the necessary preparation, your upgrade should be smooth and trouble-free.

Start the tasks in this manual after you have read the “I/O Convergence” section of the *Release Notes for HP-UX 10.10*. If you have not already retrieved the *Release Notes* from the “HP-UX Upgrade Tools” tape or CD, you can find instructions for doing so under “Locating and Loading Tools and Documentation” in Chapter 4 of this manual.

Who Should Install HP-UX 10.x?

You should normally “cold install” release 10.01 or 10.10 only if you have bought a new HP 9000 computer on which the operating system is not already installed; otherwise, you should upgrade.

Read the cautionary note on the next page if you are thinking of installing an HP-UX 10.x release onto a system already running a 9.x or earlier version of HP-UX.

Who Should Upgrade to HP-UX 10.x?

You should upgrade, as opposed to “cold-installing”, if you are already running HP-UX 9.x and you want to take advantage of the new features available in HP-UX 10.01 or 10.10 (see the *Release Notes for HP-UX 10.10* for information about these features).

What If You Are on an Earlier Release?

If you are running an HP-UX release earlier than 9.0, you must upgrade to a 9.x release before you can upgrade to a 10.x release.

Alternatively, you could “cold-install” HP-UX 10.01 or 10.10 and then re-customize your system from scratch, but, except in the case of a minimal or test system, this is probably more work than you want to do. Read the cautionary note on the next page if you are considering this option.

What If You Are Already on 10.0 or 10.01?

Updating to 10.01 or 10.10 should be a simple task. You need the manual *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

Caution

If you are currently running an HP-UX 9.x release:

The structure of the HP-UX operating system has changed as of 10.0: for example, how the file system is organized and how the system addresses peripheral devices. See the *Release Notes for HP-UX 10.10* for more information.

Because of the scope of these changes HP supports only one method of *upgrading* a system from 9.x to a 10.x release: that is, running the upgrade software as described in Chapter 6. This software will convert the file system and system configuration files for you.

If you *install* 10.x onto a system that was previously running an earlier HP-UX release, you will have to re-customize the system from scratch. If you try to merge customized 9.x system files with their 10.x counterparts, you are unlikely to succeed in producing a fully operational system, and HP will not be able to support the result.

Who Should Not Upgrade to HP-UX 10.x?

Hardware Constraints

Some systems cannot be upgraded to HP-UX 10.x under any circumstances; these include all Series 300 and 400 computers (though some 400 models can be converted to 700s).

Some Series 800 systems cannot be upgraded with an 8Mb memory card in the first slot; see Chapter 4 for more information.

Other systems will need upgrades to memory (16Mb is the minimum configuration) or disk space.

See the *Release Notes for HP-UX 10.10* for more information on which hardware is and is not supported. The `snoop` tool will tell you if your system is short of disk space; see “Analyzing Your System” in Chapter 4 for more information.

Software Constraints

Only systems running HP-UX 9.0 or later can upgrade to 10.x.

9.x to 10.10: a Two-Step Process

Because of the changes to the filesystem layout that occurred at 10.0, every system upgrading from 9.x must go through a conversion process. This conversion happens during the upgrade from 9.x to 10.01.

Upgrade to 10.01 Required

A 9.x system cannot be upgraded directly to 10.10; you *must* upgrade to 10.01 before you can update to 10.10 or any later release.

In outline, the process looks like this:

1. Prepare for upgrade:
 - a. Read *Release Notes*, white papers, etc.
 - b. Do preparation tasks.
See Chapter 4.
2. Upgrade to 10.01.
See Chapter 6.
3. If you are upgrading an HP-UX cluster, run tools to upgrade the cluster clients and convert them to NFS Diskless.

See “Converting HP-UX (“DUX”) Clusters” in Chapter 2, “Preparing for Changes to HP-UX Clusters” in Chapter 4 and “Upgrading HP-UX (“DUX”) Cluster Clients” in Chapter 6.
4. Load 10.01 applications and any optional 10.01 features you need.
See “Loading New 10.01 Features and HP Applications” in Chapter 7.
5. Update to 10.10.
See “Updating to 10.10” in Chapter 7.
6. Update applications to 10.10.

New Upgrade Capabilities

Some 9.x systems could not be upgraded to 10.01 with the tools provided with the initial release of 10.01. These systems were:

- HP-UX “DUX” clusters
- Systems running HP-UX 9.07
- Series 700 Model J systems
- Series 700 C-class workstation Models C100 and C110

These systems can all now be upgraded using the tools in the package that includes this manual. (The package is labelled “HP-UX Upgrade Tools for 9.* to 10.*”; we’ll refer to it as the “Upgrade Tools” package from here on.)

Using these tools you can upgrade your 9.x system to 10.01, or to 10.10 via 10.01. Unless you are running 9.07, you can choose whether to stop at 10.01 or go on to 10.10.

Caution **9.07:** If you are running 9.07, you *must* upgrade all the way to 10.10; you can’t stop at 10.01.

Upgrade Paths

System/Release	Upgrade and Run 10.01?	Upgrade and Run 10.10?
9.07	no	yes
9.07 “DUX” cluster	no	yes
Other 9.x S700 “DUX” cluster	yes	yes
S700 J	yes	yes
S700 C Class C100/110	yes	yes
All other S700/800 9.x	yes	yes

You can think of the above table as a train schedule: every 9.x to 10.10 upgrade goes through 10.01, but the 9.07 train does not stop there—although these systems will in fact load a version of 10.01, that version is not supported, and must not be used, for any purpose but to get the system to 10.10.

Note

As with the HP-UX operating system, HP and third party applications may also need to go through a conversion to accommodate the new filesystem layout. For this reason, if you are planning to upgrade to 10.10, you should *not* wait until you get to 10.10 to upgrade your applications.

Upgrade applications in the same way as you upgrade HP-UX: first to 10.01, then to 10.10:

1. Upgrade the system to 10.01.
2. Upgrade applications to 10.01.
3. Update the system to 10.10.
4. Update applications to 10.10.

See Chapter 7 for more information on upgrading applications distributed via **Software Distributor**, the HP utility which replaces `/etc/update`, and which you will be using to upgrade HP-UX itself. For information on analyzing and (if necessary) converting in-house code and scripts for use on 10.x, see Chapter 5.

Where To Go Next

Table 1-1.

To do this	Go to
Upgrade several systems from HP-UX 9.x	Chapter 2
Upgrade a “DUX” cluster from HP-UX 9.x	Chapter 2
Upgrade one system in a network from HP-UX 9.x	Chapter 2
Upgrade a system that is not in a network from HP-UX 9.x	Chapter 3

Note

If you need to

- Update a system from HP-UX 10.0 or later, *or*
- “Cold-install” a 10.x system

you need a different manual, *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

2

Planning a Site Upgrade

This chapter contains advice mainly intended for administrators of workgroups or larger groupings of HP 9000 computers (including HP-UX “DUX” clusters), who want to upgrade all or many of them in the same period of time.

But even if you administer only one system, you should read at least the section on “Operating 9.x and 10.x Systems Together” if your system has any network connections; and, if you have HP-UX “system” directories NFS-mounted, you need to read “Shared HP-UX “System” Directories” as well.

If none of this applies to you, you can skip this chapter and go on to Chapter 3.

Setting a Strategy

It can take something like a full shift to upgrade a single, moderately complex system. (Detailed time estimates are in Chapter 3.) This does not mean you can only upgrade one machine a day; much of the downtime is taken up by the software loading files and converting the filesystem layout and configuration files. But it does mean you should not plan to upgrade an entire site in a day—a small workgroup of fewer than ten similarly configured machines is probably the most you should plan to get done without help.

If you manage more machines than that, you will need a strategy for staggering the upgrades over a period of time, possibly an extended period of time if you need to order new hardware for some machines, or if some systems run third-party applications that have not yet been certified on 10.x.

Factors To Take into Account

Your strategy will probably need to take account of the following:

1. Need: which systems need 10.01 or 10.10 soonest.
2. Ease: which systems require the least work to upgrade.
3. Affinity: which systems are similar.
4. Interoperability: which machines depend on which others.

Need

You and your users must determine which systems need HP-UX 10.01 or 10.10 the soonest. The *Release Notes for HP-UX 10.10*, and the *Release Notes* for individual HP products, provide summaries of the new features.

Ease

The `snoop` tool, provided with the 9.x-to-10.01 upgrade software, will help you identify what, if anything, must be done before a given system can be upgraded (see Chapter 4 for details).

In general, defer upgrading systems that will require the most work, allowing time to order disks, memory or software, and to do additional administration such as removing unneeded filesets.

2-2 Planning a Site Upgrade

Affinity

Plan to upgrade similar systems at the same time. Once you have successfully upgraded the first system, you should be able to take shortcuts with the remaining systems. See “Strategy for Similarly Configured Systems” later in this chapter.

Interoperability

By **interoperability** we mean the ability to share resources such as NFS-mounted file systems and network printers. Systems that provide applications or services to a common pool of users should also be considered under this heading.

Plan to upgrade systems that require interoperability at the same time, as far as possible. If it is not possible, you will need to plan for systems still running 9.x to continue to operate with systems running 10.x. HP has provided tools to facilitate this; see “Operating 9.x and 10.x Systems Together” later in this chapter.

Shared HP-UX “System” Directories. It is possible for 9.x systems to share HP-UX “system” directories by means of NFS mounts; for example, one workstation might mount `/usr/local` from another workstation that has more disk space.

It is *not* possible to share such directories between 9.x and 10.x systems. For example, If you try to upgrade a 9.x system that has `/usr/local` NFS mounted from another 9.x system, the result will be:

- `/usr/local` on the remote system will not be touched.
- The 10.01 files that should go into `/usr/local` will not be loaded.

In this case, you should do the following:

1. Arrange to upgrade all systems sharing any 9.x “system” directory (such as directories under `/usr`) on the same day or during the same shift.
2. Shut down all of these systems at the same time.
3. Upgrade the NFS server system.

Users of this system can now log back in.

4. Upgrade the NFS client system(s).

This will unmount all the NFS-mounted directories, leaving empty directories in their place. These directories will still be empty at the end of the upgrade.

5. Remount the NFS-mounted directories on the client system(s).

Users can now log back in.

A Sample Strategy

Your site upgrade strategy might look something like this:

1. Sketch out the order in which you would ideally like to upgrade your systems, given your users' needs for 10.x features, 10.x-compatibility of critical applications, and so on.
2. Understand the 9.x-to-10.x interoperability issues, and familiarize yourself with the tools and procedures HP has supplied to help solve them (see "Operating 9.x and 10.x Systems Together", later in this chapter).
3. Choose "typical" systems and analyze them using the tools supplied by HP (see Chapter 4).

By "typical", we mean a system that is similar to a number of other systems in hardware type (Series 700 or 800), disk space, and peripheral and networking configuration. (If you have a number of systems that are *very* similar, you may be able to use the procedure suggested under "Strategy for Similarly Configured Systems" later in this chapter.)

4. Use this analysis to pre-qualify the remaining systems on your list.

You should be able to divide the list roughly into three categories:

- a. Systems that can be upgraded with minimal preparation and post-upgrade effort.

These would be systems on which all of the following are true:

- The **snoop** tool reports no **PROBLEMS** (see Chapter 4).
- Critical third-party binary applications are certified for 10.x.

Most 9.x applications should run without problems on 10.x, because of **transition links** that link 9.x HP-UX pathnames to their 10.x equivalents (see Chapter 5 for more information.) But, to be certain,

2-4 Planning a Site Upgrade

you should contact application suppliers, and you may also want to pre-qualify critical applications on a 10.x test system.

- The **prepare** tool reports few compatibility problems in source code and scripts, other than HP-UX pathname differences (which **transition links** handle). See Chapter 5.

b. Cases that will take more time.

These would include systems that need more disk space, or need hardware upgrades, or form part of an HP-UX cluster (see “Converting HP-UX (“DUX”) Clusters”, later in this chapter); or that run critical code or scripts that require modification for 10.x (see Chapter 5 and Chapter 8).

c. Systems that can’t be upgraded, for example Series 300 systems (see Chapter 3).

5. Rework your list so that systems near the top have the following characteristics:

- Need to run 10.x as soon as possible.
- Present few or no problems for upgrade.
- Run code and scripts that require little conversion effort.
- Are typical of a number of other systems.

We’ll refer to these as “good candidates” from now on.

6. Resolve any configuration problems the tools report on the “good candidate” systems (see Chapter 4).

7. Decide on your strategy for converting source code and scripts on the “good candidates” (see Chapter 3).

8. Convert source code and scripts on the “good candidates” if you so decide (see Chapter 4 and Chapter 5).

Pay particular attention to code or scripts that are re-used on several systems; it may be a good investment of time to convert these now. Users’ `.login` and `.profile` scripts may also need individual attention if they are heavily customized.

9. Upgrade the “good candidate” systems to 10.01 (see Chapter 6).

10. Convert startup and shutdown scripts, and do other post-upgrade tasks, on the “good candidates” (see Chapter 7).
11. Upgrade applications on the “good candidates”, and update them to 10.10 if you so decide.

Note You *must* upgrade systems running 9.07 to 10.10. You will initially upgrade these systems to an intermediate version of 10.01, but that version is supported *only* for the purposes of upgrading it immediately to 10.10.

12. Implement 9x-to-10.x interoperability strategies on systems that cannot be, or will not be, upgraded soon (see “Operating 9.x and 10.x Systems Together”, later in this chapter).
13. Analyze the remaining systems (see Chapter 4).
14. Upgrade the remaining systems (see Chapter 6).
15. Convert startup and shutdown scripts on the remaining systems (see Chapter 7), or copy these scripts from a similar system that has already been upgraded.
16. Upgrade applications on these systems, and upgrade them to 10.10 if you so decide (see Chapter 7).

Strategy for Similarly Configured Systems

If you have several systems that are much alike in their configuration and are all used for the same or similar purposes, you may be able to simplify the procedure outlined above.

For example, you might have a workgroup of Series 750s, all running HP-UX 9.05 and the same version of the most commonly used or mission-critical applications from a single root disk and single LAN card, with similar networking and I/O configuration.

For such a group of similar systems, you can do the following:

1. Run **snoop** to analyze one system (see “Running snoop” in Chapter 4).
2. Fix any problems **snoop** reports (see “Modifying Your System” in Chapter 4).
3. Analyze source code, **.login** and **.profile** files and other user scripts, and convert them if necessary (see Chapter 3 and Chapter 5).
4. Upgrade this system to 10.01 (see Chapter 6).
5. Modify system startup and shutdown scripts and do any other post-upgrade tasks that may be necessary (see Chapter 7).
6. Install and test converted 9.x code and scripts if necessary.
7. Qualify 9.x application binaries using your normal regression-testing procedures.
8. Upgrade HP applications (and third-party applications for which a 10.x upgrade is available), and update this system to 10.10 if you so decide.
9. Fix any problems on the remaining systems that **snoop** warned you about when you ran it on the first system.
10. Run **snoop** in unattended mode on each of the remaining systems (see “Modifying Your System” in Chapter 4 for an example).

Check the **snoop.log** file on each system to make sure there are no **PROBLEMS**, and no **CAUTIONS** you do not already know about from the first system.

11. Run **upgrade** in unattended mode on each of the remaining systems.
Make sure you check the logfiles before allowing users back on (see “Examining the Upgrade Log Files” in Chapter 7).
12. Copy over code and scripts from the first system to each of the remaining systems, adapting them to the individual systems and re-testing them as needed.
13. Copy over application binaries certified on the first system.
14. Upgrade HP applications (and third-party applications for which a 10.x upgrade is available), and update the remaining systems to 10.10 if you so decide.

Sharing Disk Space

If you have one large Series 700 or 800 system (with plenty of spare disk space and memory), and one or more low-end Series 700 systems, you may want to consider implementing **NFS Diskless**.

You can save disk space by upgrading the large system to 10.01, configuring it as an NFS Diskless server, then booting the remaining systems as NFS Diskless clients, freeing up space on the clients' local disks (you can now remove the HP-UX 9.x files from these disks).

See the 10.01 version of the *HP-UX System Administration Tasks* manual for information on configuring and managing NFS Diskless; for a high-level overview, see the *Release Notes for HP-UX 10.10*. (Instructions for getting the *Release Notes for HP-UX 10.10* from the 10.01 upgrade tape or CDROM are under “Locating and Loading Tools and Documentation” in Chapter 4.)

Note If you are currently running an HP-UX cluster, see “Converting HP-UX (“DUX”) Clusters” later in this chapter.

Operating 9.x and 10.x Systems Together

HP has provided tools and procedures to help you to continue to operate 9.x and 10.x systems together, sharing peripherals, NFS mounts, etc., as you did when all the systems were running 9.x. For more information, see the *HP-UX 9.x/10.x Interoperability Guide* (HP part number 5963-8920), supplied in the same “Upgrade Tools” package that includes the manual you are reading.

HP-UX 10.x systems lend themselves particularly well to operating with systems running other versions of HP-UX. For example, you can mount “non-system” directories (directories that do not form part of HP-UX itself) from 9.x systems onto a 10.x system; you can do this either by means of NFS mounts, or you can actually move non-system disks from HP-UX 9.x to 10.x systems (but Series 700 Software Disk Striping is not supported on 10.x; see “Considerations for Series 700s” in Chapter 4). In addition, if you configure NFS Diskless on the 10.x system, you can swap dynamically to any system, including a 9.x system.

These characteristics mean not only that 10.x systems communicate easily with 9.x systems, but also that you may be able to save disk space on your 10.x systems by sharing “non-system” 9.x disks.

10.01 User Login Scripts

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

For more information, see Chapter 6 of the *HP-UX 9.x/10.x Interoperability Guide*.

Converting HP-UX (“DUX”) Clusters

HP-UX (“DUX”) clusters are no longer supported as of HP-UX 10.0.

The package that includes this manual provides tools that convert clusters of Series 700 computers from HP’s proprietary technology, “DUX”, to the new technology introduced at 10.01, which is based on NFS Diskless.

Note

- HP recommends that you use these tools to upgrade Series 700 “DUX” clusters. This is the only method HP supports for upgrading a “DUX” cluster to 10.x.
- No Series 300 or 400 computer can be a member of an NFS Diskless cluster, nor can these computers be upgraded “standalone” to HP-UX 10.x.

See “Providing for Series 300 and 400 Clients”, later in this chapter.

Upgrading a “DUX” Cluster

As with a standalone system, you must upgrade a cluster to 10.01 before you can update it 10.10.

In outline, the steps for upgrading a “DUX” cluster from 9.x are as follows. You’ll find detailed procedures later in this chapter, and in chapters 4, 6 and 7, as noted below.

1. Remove all Series 300 and 400 clients from the cluster.
See “Providing for Series 300 and 400 Clients”, later in this chapter.
2. Remove all inactive clients and client “fragments”.
“Inactive clients” means clients that you never intend to boot to the new NFS Diskless cluster; “fragments” are kernels or device or configuration files for clients that are no longer part of the cluster.
See “Removing Series 300/400 Clients and Inactive Clients” in Chapter 4.
3. Run the analysis tools on the cluster server.
See “Analyzing Your System” in Chapter 4.
4. Upgrade the server to 10.01.
See Chapter 6.
5. Upgrade the clients.
See “Upgrading HP-UX (“DUX”) Cluster Clients” in Chapter 6.
6. Upgrade applications, and update the cluster to 10.10 if you so decide.
See Chapter 7.

What Does the Conversion Do for You?

The “DUX” upgrade tools convert a running 9.x HP-UX cluster to a 10.01 NFS Diskless cluster. As far as possible, the conversion happens automatically, restoring “DUX” functions (or their nearest equivalents) on the NFS Diskless clients, and requiring you to intervene only when decisions have to be made.

The tools expect all cluster clients to be booted to the cluster server at the time of the upgrade (and of the pre-upgrade analysis). See “Running snoop” in Chapter 4 and Chapter 6. The table that follows shows what “DUX” functions will be restored on these clients when they come up on 10.x under NFS Diskless.

For information on what happens to clients that are not booted to the server, see “If Clients Are Not Booted”, later in this chapter.

"DUX" Function	Restored on 10.01?	"DUX" Function	Restored on 10.01?
=====			
FILE SYSTEMS:		SPOOLING:	
- on server's disks:	no*	- to printers attached to server:	yes
- on client's own disks:	yes	- to printers attached to this client:	yes
- on another client's disks:	no**	- to printers attached to another client:	yes
SWAP:		- to network printers:	yes
- to root server:	yes	SYSTEM CONFIG INFO:	
- to local disk:	yes	- networking	yes
- to swap server (device swap to another client's disk):	no	- critical config files (inittab, fstab..)	yes
- to file server (f/s swap to another client's disk):	yes	DEVICE FILE INFO:	yes
BOOT FROM SERVER:	yes	EMAIL:	yes
KERNEL:	yes	CLUSTER-WIDE LOGINS:	yes
		APPLICATIONS	no~

* = HP-UX "system" directories only; use SAM to export others
** = Use SAM to NFS-mount these file systems
~ = See "Applications" later in this chapter

The subsections that follow provide more information about the conversion.

File Systems. In a 9.x “DUX” cluster, all file systems on the server are automatically visible to the clients, and a file system resident on a client’s local disk is visible to the server and all the other client.

This is not the case under NFS Diskless. Directories that comprise the HP-UX operating system, and directories such as `/export/private_roots/client` that contain the client’s configuration files (and, by default, swap files) are automatically exported, but other directories on the server’s and clients’ disks are not.

- The following directories *are* automatically exported from the server and mounted on all clients:

- `/sbin`
- `/usr`
- `/opt/*` *if optional products are configured for diskless*
- `/home` *if policies specify shared home dirs*
- `/var/mail` *if policies specify shared mail*

For information on “shared” versus “private” cluster policies, see the *NFS Diskless Concepts and Administration* white paper; “Locating and Loading Tools and Documentation” in Chapter 4 explains how to get it off the tape or disk that is packaged with this manual.

- Other directories *are not* automatically exported and mounted; you need to do this explicitly. SAM, the menu-driven System Administration Manager simplifies the task. See “Procedure for Upgrading Clients” in Chapter 6 for more information.

See Appendix B for more information on managing file systems and disk space in an NFS Diskless cluster.

Swap. By default, NFS Diskless clients swap to their `/paging` directory, which is in the server's disk space (`/paging` on a client is `/export/private_roots/client/paging` on the server). But, as far as possible, the 9.x-to-10.01 upgrade preserves the swapping arrangements used by your 9.x "DUX" cluster.

For clients that are booted to the server during the upgrade, the upgrade program converts DUX swap configurations to NFSD as follows:

"DUX" Configuration	Converted to NFSD Configuration
Device swap to server	File-system swap to server
Device swap to local disk	Device swap to local disk
Device swap to another client	File-system swap to server

In addition, file-system swap configured under "DUX" will be restored under NFS Diskless.

Caution

Do not configure an NFS Diskless client to swap *only* to another client's file system.

Configure primary swap to the server or a local disk; otherwise the client may fail to boot.

Kernel-Configured Swap. Under 9.x, swap to a client's local disk can be configured via a statement in the client's `dfile`; on NFS Diskless clients it cannot.

If the client is booted to the server during the upgrade, the upgrade process will convert local device swap configured in the `dfile` to local device swap configured in `/etc/fstab` (the 10.x equivalent of `/etc/checklist`).

Note

- If you have configured local swap both in the `dfile` and in `/etc/checklist`, the upgrade process will choose the `checklist` version (whereas 9.x HP-UX uses the `dfile` version).
 - If you have configured swap by means of a command-line `swapon` statement only, and have not put a swap entry in either the `dfile` or `/etc/checklist`, this client will get default swap (file-system swap to the server).
-

Too Much Device Swap? By default, NFS Diskless clients swap to the server's *file system*, whereas under "DUX" they swap by default to the server's *device swap* space (**device swap** refers either to a swap area at the end of a file system, or to a disk dedicated to swap that has no file systems on it).

In fact, NFS Diskless clients *cannot* use device swap on the server, or any other remote system, but only on their own local disks. (NFS Diskless clients *can* use NFS file-system swap on a remote system, whether or not that system is part of the cluster, and whether or not it is running HP-UX 10.x.)

Because of these differences, you may have too much device swap configured on the server's disks, and not enough room for file-system swap.

For 9.x "DUX" clusters, HP recommended 30Mb device swap per client as a rule of thumb. Under NFS Diskless, HP recommends a minimum of 44Mb file-system space per client for all purposes, including kernel, boot files and file-system swap. By default, this space is used under `/export` on the server (but you can allocate the space elsewhere and link it to directories under `/export`, as explained in Appendix B).

You should probably allow for this much space whether or not clients are configured to swap to their local disks, because all clients swap to the server while they are booting.

Note When you run the `snoop` analysis tool, described in Chapter 4, and `upgrade`, described in Chapter 6, the tools will do a disk-space analysis on the basis of the software you select. In the case of a cluster server, this analysis assumes that all the clients will be swapping to the server's disk space (always true when they are booting).

This conservative analysis may show a surprising increase over 9.x in the amount of disk space required, but it ensures that all the clients will be able to boot at the same time (for example, when you first bring them up as NFS Diskless clients).

Remedies:

- The `snoop` and `upgrade` tools allow you to reclaim device swap that is at the end of the file system space on the server's disks.

`snoop` will warn you if you appear to have more device swap on the server than you will need, and at that point, *before* you run `upgrade`, you can create a file that will be read during the upgrade process; `upgrade` will reclaim the amount of swap you specify, on the disks you specify, and add it back to the file system.

Caution This will be your only chance to reclaim this space without doing a full backup of the data on the disk.

See “Reclaiming Device Swap” in Chapter 4 and “Procedure for Upgrading Clients” in Chapter 6 for more information.

- You can use ordinary HP-UX commands to create a file system on a disk that was formerly dedicated to device swap.

You should probably do this either just before or just after upgrading the server to 10.01, so as to keep your “DUX” cluster up and running as long as possible, but in time to make server swap space available before you boot the clients under NFS Diskless.

See Appendix B for more information on managing disk space and swap in an NFS Diskless cluster.

Applications. Not all HP and third-party applications are certified to run on NFS Diskless clusters. **Contact the suppliers of critical applications before you commit to upgrading a cluster to 10.x.**

If Clients Are Not Booted. The tools will provide defaults for clients that are not booted to the cluster server while it is being upgraded, but, as the table on the next page shows, you will need to do more work to recapture the 9.x configuration of these clients; they will look to the new NFS Diskless cluster much like new clients, lacking information about peripherals, users, etc. (The **snoop** analysis tool, discussed in Chapter 4, refers to such clients as “offline” clients.)

"DUX" Function	Restored on 10.01?	"DUX" Function	Restored on 10.01?
=====			
FILE SYSTEMS:		SPOOLING:	
- on server's disks:	no*	- to printers attached to server:	yes
- on client's own disks:	no%	- to printers attached to this client:	no
- on another client's disks:	no**	- to printers attached to another client:	no
SWAP:		- to network printers:	no
- to root server:	yes	SYSTEM CONFIG INFO:	
- to local disk:	no%	- networking	no%
- to swap server:	no%	- critical config files (inittab, fstab..)	no%
- to file server:	no%	DEVICE FILE INFO:	no%
BOOT FROM SERVER:	yes	APPLICATIONS	no~
KERNEL:	no%		
CLUSTER-WIDE LOGINS:	yes		
EMAIL:	yes		

* = HP-UX "system" directories only; use SAM to export others
** = Use SAM to NFS-mount these file systems
% = Defaults similar to new NFS Diskless clients;
all attached peripherals must be reconfigured
~ = See "Applications" earlier in this chapter

Clients with “Dual Personalities”. HP-UX 10.x releases are designed to work well with previous HP-UX releases. This means, for example, that if a client is a member of two “DUX” clusters, you can upgrade one cluster to 10.x, boot the client to the 10.x cluster, and from then on boot the client to either server, whether the servers are running 8.x and 10.x, 9.x and 10.x, or 10.01 and 10.10.

Similarly, 10.x clients will be able to boot standalone from a local disk containing any supported O/S.

Of course, if you reconfigure the client’s disks, you may destroy or compromise the client’s “other personality”. For example, if you configure a 10.x client’s local disk for LVM, then boot the client to a 9.x cluster, the 9.x client will not be able to use the data on the disk, and must not swap to it.

Managing Root’s Home Directory

In HP-UX “DUX” clusters, all directories are visible on all systems in the cluster, so the root user’s home directory is automatically available on all cluster nodes. This is not true for NFS: the root user’s home directory is *not* automatically shared on the clients (whether or not `root`’s home directory is `/`).

You could force the root user’s home directory to be shared, for example by making it a subdirectory of a directory that *is* shared automatically, such `/home`, but HP recommends the following:

- Make root’s home directory `/` (as it is by default).
- Never try to share the `/` directory.
- Limit the files in this directory to the essential “dot” files (`.profile`, etc.)
- Put any scripts or applications you want to reserve for the root user in some other directory in the root user’s path, and export the directory to the clients.

See “Procedure for Upgrading Clients” in Chapter 6 for more information on sharing directories.

Providing for Series 300 and 400 Clients

Because 10.x does not support Series 300 and 400 clients, you must remove any such clients from a Series 700 cluster *before you upgrade the server* to any 10.x release. Before you do that, though, you will need to make some other provision for these clients. (The `snoop` tool, discussed in Chapter 4, refers to these as “68K clients”, as shorthand for their 680n0 processor architecture.)

Here are some of the options:

- Convert the clients to standalone machines.
- Convert one client to be a Series 300/400 cluster server.
- Upgrade clients to Series 700s.
- Convert clients to Xterminals.

Each of these options is discussed in the bullets that follow.

- Convert the clients to standalone machines.

If the clients have at least one disk each, you can install HP-UX onto each client. Follow directions in the Series 300/400 version of *Installing and Updating HP-UX 9.0*.

To avoid re-certifying applications, install a version of HP-UX that is compatible with the version currently installed on the Series 700 server (for example, if the server is running 9.03, install the Series 300/400 version of 9.0 or 9.03 on the former clients).

The Series 300/400 systems' disks *do not* each need to have a copy of all the non-system files stored on the Series 700's disks; instead, they can NFS-mount the directories they need from the Series 700.

Follow the directions for doing NFS mounts in chapter 6, “Managing the File System,” of the 9.0 Series 300/400 version of the *System Administration Tasks* manual (HP part number B1864-90010).

If you choose this option, you may need to buy disks, and you will lose the advantage of “single-point administration.”

- Convert one client to be a Series 300/400 cluster server.

A Series 300 or 400 cluster server needs at least 16 Mb of memory and at least a 420 Mb disk—enough disk space for HP-UX 9.0, and an additional 30 Mb per client for swap, unless the clients can swap to their own or another client's disk.

See the 9.0 Series 300/400 version of *Managing Clusters of HP 9000 Computers* (HP part number B1864-9015) for instructions on configuring a cluster server and details of hardware requirements.

Once you have configured the server, add the other Series 300/400 machines as clients, and NFS mount non-system directories from the Series 700 server (see previous bullet).

If you choose this option, you may need to buy more memory or a bigger disk for the Series 300/400 server, and you will have to manage two clusters where there was only one before, but this may be the least expensive and most efficient of the options.

- Upgrade clients to Series 700s.

If all of your clients are Series 400s, you may be able to buy board upgrades to convert them to Series 700s. Board upgrades are not available for all models of Series 400; check with your HP Sales Representative (SR).

If it turns out that you can convert all of the Series 400 clients to 700s, do the following to upgrade the cluster to 10.01:

- Remove the Series 400 clients from the cluster.
- Do the hardware upgrade to convert them to 700s.
- Upgrade the cluster to HP-UX 10.01, converting it to NFS Diskless using the special tools provided by HP.
- Upgrade the cluster to 10.10 if you so decide.
- Add back the clients that you converted to 700s.

(Alternatively, once you've done the hardware conversion, you can add the clients back as Series 700 "DUX" clients under 9.x, then upgrade and convert the cluster to NFS Diskless.)

This option leaves your cluster intact and frees up disk space (you now only need the Series 700 version of HP-UX), but is not available for Series 300s and only for certain Series 400 models.

2-22 Planning a Site Upgrade

- Convert Series 300 and 400 clients to Xterminals.

Xterminals are graphics-display terminals; if your clients are truly running “diskless” (without any disks of their own) and are not supporting local printers or other peripherals, then in effect you are already using them as terminals, and converting them to Xterminals may be your best option.

To convert Series 300 and 400 clients to Xterminals, you will need the XTERM300 product which is part of HP-UX 9.10 (XTERM300 cannot be ordered separately); consult your HP Sales Representative for more information.

For more information on this option, see the section “Using the Series 300/400 Xterminal Product” in chapter 5 of the *HP-UX 9.x/10.x Interoperability Guide*.

Summary of “DUX” Issues

- HP-UX “DUX” clusters are no longer supported as of 10.0.
- 10.01 introduces a new kind of HP-UX cluster, HP’s much enhanced version of NFS Diskless.
- The package of which this manual is a part includes special tools to upgrade and convert 9.x “DUX” clusters of Series 700 computers to 10.01 NFS Diskless clusters.
- No Series 300 or 400 system can be a member of an NFS Diskless cluster.
You will have to make other provisions for these clients; see “Providing for Series 300 and 400 Clients” earlier in this chapter. You must remove these clients from the Series 700 cluster before you run the upgrade.
- Not all HP and third-party applications will work correctly on all members of a cluster; if in doubt, contact the supplier of the application.

Planning a System Upgrade

This chapter explains what you need to do to prepare to upgrade a system to 10.01. Check the bullets that follow to make sure you're in the right place.

- If you are running a release earlier than 9.0, do not go on.

You either need to upgrade your system to a 9.x release first, or, possibly, “cold install” 10.01 and reconfigure your system from scratch (as though it were a new system):

- For “cold install” instructions, see *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.
- For instructions for upgrading from 8.x to 9.x releases of HP-UX, see *Installing and Updating HP-UX 9.0*.
- If you are running HP-UX 10.0 or 10.01, you do not need to do the tasks described in this manual; see *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

The section that begins on the next page provides information on systems you cannot upgrade to 10.x (or which will require hardware modifications before you can upgrade them). If none of this applies to your system, go on to the next section of this chapter, “Upgrade Paths”.

Who Should Not Upgrade

Systems that Cannot Be Upgraded

The following systems cannot upgrade to HP-UX 10.01:

- Series 300 and 400 workstations (including “DUX” clients of Series 700 cluster servers).

No Series 300 or 400 computers are supported on 10.x. Some Series 400s can be converted to Series 700s by means of a hardware upgrade; consult your HP representative if you want to explore this option.

- Some older Series 600, 700 and 800 computers.
- Systems with less than 16Mb of RAM.

An NFS Diskless cluster server must have a minimum of 32Mb.

- Some Series 800s with an 8Mb memory card in the first slot.

The `snoop` tool will warn you if your system may have this problem; see Chapter 4 for more information.

See the *Release Notes for HP-UX 10.10* for more details.

Upgrade Paths

Upgrade Tools

10.x involves major changes both to the structure and content of HP-UX. To smooth the transition from 9.x, HP has provided tools to:

- Automate the conversion of the file system and system configuration files.

These tools are invoked automatically during the upgrade itself; see Chapter 6 for more information.

- Link 9.x system pathnames to 10.x system pathnames.

These links, called **transition links**, will be installed by default when you upgrade to 10.01. They will allow many 9.x applications to run without modification on 10.01.

- Analyze 9.x source code and scripts for incompatibilities with 10.x, and optionally modify system pathnames.

Chapter 8 provides details of the potential issues; they arise mainly from:

- Changes to system pathnames.
- Changes to commands.
- Changes to system and library calls.

The tools will help you identify potential problems in source code, scripts and documentation.

Caution These tools provide information that is valid for 10.01, but do not reflect changes between 10.01 and 10.10.

- Analyze your 9.x system for any configuration or other problems that could cause the upgrade to fail.

For example, you might not have enough disk space or memory, or you might be running optional software that is not supported on 10.01, such as DataPair/800.

The **snoop** tool, described in Chapter 4, will give you precise information about each system you run it on.

Upgrade Options

There are two possible approaches you should keep in mind when planning your upgrade, though you may in fact decide to combine the two.

- Option A: Quick Start
- Option B: Full Preparation

The following subsections provide details of each approach.

Note Bear in mind as you read that if you are going to 10.10 you must do a two-stage upgrade: first to 10.01, then to 10.10. See Chapter 1 for an outline of the whole process.

Option A: Quick Start

This approach relies on the **transition links** (sometimes referred to as “fast transition links” or “compatibility links”). Transition links are installed by default on all 10.x systems; they link HP-UX 9.x “system” pathnames to their 10.x equivalents, and they are intended to reduce the amount of conversion work you will have to do on 9.x code and scripts when you first come up on a 10.x system.

If you adopt this approach entirely, you will simply run your normal regression tests on applications and scripts once you come up on 10.01 or 10.10, just as you would after any other upgrade. You should find that much of your code runs without problems.

But do not rely on transition links to solve all problems, or as more than a short-term aid; HP does not intend to support them beyond 10.x and they will only help with problems related to pathnames; there could be other problems.

For example, if your code uses commands or system calls that are no longer supported or whose options or behavior have changed, or if it builds or makes reference to 9.x device files (other than “default” names such as `/dev/lp` which are still recognized by HP-UX), you will have to fix it.

HP recommends that you do the following for a “quick start”:

3-4 Planning a System Upgrade

1. Read Chapter 8 of this manual to get an idea of the potential problems.

You should also read Chapter 3 of the *Release Notes for HP-UX 10.10* to understand the major changes introduced into HP-UX at 10.0/10.01. (See “Locating and Loading Tools and Documentation” in Chapter 4 for instructions for getting the *Release Notes for HP-UX 10.10* from the upgrade tape or CD).

2. Do the preparation tasks described in Chapter 4.
3. Run the analysis tools described in Chapter 5, but don't worry about HP-UX system pathnames.

Just fix any problems with commands and system calls that the tools report.

Caution These tools *will not* catch changes that have occurred since 10.01.

4. Upgrade to 10.01
5. Do the post-upgrade tasks described in Chapter 7.
6. Upgrade to 10.10 if you so decide.
7. Over the lifetime of 10.x, do the conversion tasks described in Chapter 5.

Transition links will not be supported beyond 10.x.

For more information on transition links, see the `tlinstall(update_aid)` manpage, which you'll find on your system after you upgrade to 10.01.

Time Needed for Option A.

Estimated time required for this approach:

Pre-Upgrade:	8 - 20 hrs
Upgrade (Downtime):	4 - 12 hrs
Post-Upgrade:	3 - 5 hrs

Total:	15 - 37 hrs

Notes:

1. If you are proceeding to 10.10 after upgrading to 10.01, add two to four hours downtime for the 10.10 upgrade.
2. If you are upgrading an HP-UX (“DUX”) cluster, add about 15 minutes per client downtime to the total.

If your cluster includes Series 300 or 400 clients, you should probably figure in something like a day’s worth of pre-upgrade time making other provision for these clients; see “Providing for Series 300 and 400 Clients” in Chapter 2.

Option B: Full Preparation

This approach requires more preparation time than option A, described in the previous subsection; but it has the following advantages:

- Your 9.x applications and scripts are more likely to work first time and without problems when you first come up on 10.01, and will be more easily portable to other vendors’ “V.4” systems.

All the same, contact the suppliers of critical applications, including HP applications, well before you commit to an upgrade, to make sure these applications are supported on 10.x.

- You will have a more thorough understanding of 10.01/10.10 and how they differ from earlier HP-UX releases.
- By the time you come up on 10.01, you will have done the conversion work that must be done sometime over the lifetime of 10.x.

This approach involves the following steps:

3-6 Planning a System Upgrade

1. Read Chapter 8 to get an idea of the potential problems.

You should also read Chapter 3 of the *Release Notes for HP-UX 10.10* to understand the major changes introduced into HP-UX as of 10.0/10.01, and the additional features being added for 10.10.

2. Do the preparation tasks described in Chapter 4.
3. Run the analysis tools described in Chapter 5, and do the conversion tasks described in that chapter.

Caution If you are going to 10.10, the tools will catch most, but not all, of the changes you will need to make. They will *not* catch changes made to commands and libraries between 10.01 and 10.10.

4. Upgrade to 10.01.
5. Do the post-upgrade tasks described in Chapter 7.
6. Upgrade to 10.10 if you so decide.

Time Needed for Option B.

Estimated time required for this approach:

Pre-Upgrade:	11 - 28+ hrs
Upgrade (Downtime):	5 - 12 hrs
Post-Upgrade:	5 - 8 hrs

Total:	21 - 48+ hrs

Notes:

1. If you are proceeding to 10.10 after upgrading to to 10.01, add two to four hours downtime for the 10.10 upgrade.
2. If you are upgrading an HP-UX (“DUX”) cluster, add about 15 minutes per client downtime to the total.

If your cluster includes Series 300 or 400 clients, you should probably figure in something like a day’s worth of pre-upgrade time making other provision for these clients; see “Providing for Series 300 and 400 Clients” in Chapter 2.

Will You Need To “Re-Install?”

Re-install means installing HP-UX onto a system that is already running HP-UX. There is one case that requires you to re-install either 9.x (the version you are currently running) or 10.01/10.10 (after you have completed the upgrade): this is if you want to convert your root disk to Logical Volume Manager (LVM).

LVM is the disk management scheme HP recommends for 10.x systems, on both the Series 800 (on which it was available as of 9.0) and the Series 700 (on which it was new for 10.0).

But LVM is *not* required for upgrade to 10.x, which continues to support “whole disk” access (as on 9.x Series 700) and hard partitions (as on 9.x Series 800, for disks that supported partitions as of HP-UX 9.04).

If you intend to convert your root disk to LVM, you should factor in additional downtime. This downtime does not need to coincide with the upgrade itself; the conversion *cannot* be done while the upgrade is in progress.

The options are:

- For Series 800:
 - Re-install 9.x, converting the root disk to LVM, then upgrade to 10.01; *or*:
 - Upgrade to 10.01/10.10, then re-install 10.01/10.10, converting the root disk to LVM.

See “If You Want to Convert Hard-Partitioned Disks to LVM (Series 800)” in Chapter 4 for more information.

- For Series 700 currently running 9.07:
 - Upgrade to 10.10 (via transitional version of 10.01) then re-install 10.10.
- For a Series 700 currently running any other 9.x release:
 - Upgrade to 10.01/10.10, then re-install 10.01/10.10, converting the root disk to LVM (no other option).

Pre-Upgrade Tasks for All 9.x Systems

This chapter describes tasks that everyone upgrading a 9.x system to 10.01 or 10.10 needs to do, *well ahead of the upgrade*. See the table of tasks and times later in this chapter for lead-time estimates.

You *do not* need to do the tasks in this chapter if you:

- Are already running a 10.x release and want to go the next one.
- Are planning to “cold install” HP-UX 10.01 or 10.10.

“Cold install” means putting the HP-UX operating system onto a new computer, or onto an existing system whose configuration you don’t need to preserve (for example a “crash and burn” test system).

In these cases, you need a different manual, *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

Note The only way to upgrade a 9.x system to 10.x is to upgrade initially to 10.01. This chapter describes the steps to prepare your system for that upgrade, and Chapter 6 describes the 10.01 upgrade itself. From there it is a simple step to upgrade to 10.10. See “Summary of Upgrade Tasks From Here On”, on the next page, for an outline of the procedure.

Summary of Upgrade Tasks From Here On

The following is list of the tasks you need to do to get your system up and running on 10.01 or 10.10.

1. Load the pre-upgrade tools and documentation onto your system.
See “Locating and Loading Tools and Documentation” later in this chapter.
2. Understand the major changes from 9.x to 10.x.
See “Understanding the Changes” later in this chapter.
3. Change existing procedures.
See “Changing Existing Procedures” later in this chapter.
4. Analyze your 9.x system to make sure the upgrade will succeed smoothly.
See “Analyzing Your System” later in this chapter.
5. Make any modifications to your system that are needed for the upgrade to succeed.
See “Modifying Your System” later in this chapter.
6. Back up your system.
See “Backing Up Your System” later in this chapter.
7. Perform the upgrade to 10.01.
See Chapter 6.
8. Fine-tune the 10.01 system and upgrade applications to 10.01.
See Chapter 7.
9. Upgrade to 10.10 if you so decide (see Chapter 7).

Time Frame

You should begin the tasks described in this chapter as soon as possible, ideally three to four months before you plan to upgrade the system. This is not because the tasks themselves will necessarily take a great deal of your time, but to give yourself lead time to:

- Order and install new hardware if you need to.
- Modify pathnames (if you decide to do so) and command references, etc., in scripts and applications.
- Make provision for interoperating with systems that you cannot (or decide not to) upgrade to 10.x (see Chapter 2).

Note

DO NOT wait until you are ready to upgrade before starting these tasks; don't expect to get all the preparation work done the same shift or weekend you do the upgrade. The pre-upgrade tools are designed so you can use them over a period of time, while the 9.x system is up and running, without inconveniencing your users or interfering with the operation of the system.

The upgrade itself, on the other hand, must be done while the system is down. If you complete the procedures in this chapter before you start the upgrade, you should have your system back on line and fully operational in the minimum amount of time; if you do not, the upgrade may not succeed, and, if it does, the system is unlikely to function smoothly without considerable intervention on your part.

The table that follows suggests a timetable.

Timetable for Pre-Upgrade Tasks

The table that follows provides rule-of-thumb guidelines for when to start each task (how many weeks ahead of when you plan to upgrade), roughly how much time you can expect to elapse between the start and the end of the task (including time when a program is running and you can be doing other things) and roughly how many hours of your undivided attention you should allow, in addition to the time you'll need for the actual upgrade.

Table 4-1.

Task	Start (weeks ahead)	Elapsed time	Your time (hours)
Load tools	16-20	0.5 hours	0.25
Read documentation	16-20	2-6 hours	2-6
Make procedural changes	16-20	1-4 hours	1-4
Analyze 9.x system	16-20	0.5-several hours	0.5-3
File system preparation	12-18	1 hour-many weeks	1-24
I/O-convergence preparation	12-18	1 hour-several weeks	0-8
“DUX” cluster preparation	12-18	1 hour-several days	0-8
Recheck system	2-4	0.5-3 hours	0.5-3

Note The times in the table apply to a “system,” meaning a single machine. If you manage a group of systems that are similarly configured, the times apply to the first system you do the tasks on; for the remaining systems, the time you need to spend should be drastically reduced. For HP-UX (“DUX”) clusters, the times apply to the cluster as a whole, and, in general, you should allow more time for larger clusters than smaller ones.

4-4 Pre-Upgrade Tasks for All 9.x Systems

Locating and Loading Tools and Documentation

Loading the analysis tools and documentation files onto your 9.x system *will not* alter your system, but you will need sufficient disk space. The following table shows approximate sizes of the filesets in the tools package, which is labelled “Upgrade Tools”. `/etc/update` will show you the exact sizes of the filesets.

Table 4-2. Fileset Information

Content	Fileset	Size (Kb)
Analysis (pre-upgrade) tools	UPG-ANALYSIS	4304
Analysis tools manpages	UPG-MAN	58
Japanese manpages	UPG-JPN-E-MAN	56
Japanese manpages	UPG-JPN-S-MAN	56
<code>snoop</code> tools	UPG-SNOOP	2743
<code>upgrade</code> tools	UPG-UPDATE	2607
<code>upgrade</code> commands	UPG-CMDS	6265
Release notes, white papers	UPG-DOC	941
Software Distributor program	SD-AGENT	5251
Software Distributor commands	SD-CMDS	15833
Software Distributor text	SD-DATA	244
Software Distributor manpages	SD-MAN	220

Notes

1. These filesets include not only the tools you will need to do the pre-upgrade tasks described in this chapter, but also the tools for the upgrade itself.
 2. Load *all* of the filesets. Load the tools onto a disk on the system that is to be upgraded; the upgrade will not work if the tools are NFS-mounted.
 3. The majority of the space needed by the tools (about 27Mb) is under `/usr`.
-

The following table shows the names of the online documents that were shipped in the “Upgrade Tools” package, and the filename, size, and format of each. All the files will be installed into `/etc/newconfig/10RelNotes` on a 9.x system. These documents were not shipped to you in hard copy; to get a hard copy, use the retrieval and printing instructions that follow the table.

Table 4-3. Online Documents Information

Document	Size (Kb)	Format	Filename
<i>Release Notes for HP-UX 10.10</i>	638	ASCII	<code>10.10RelNotes</code>
<i>HP-UX 10.0 File System Layout</i>	208 81	PostScript ASCII	<code>filesys.ps</code> <code>filesys.txt</code>
<i>HP-UX 10.01 Patch List</i>	55	ASCII	<code>sw_patches.1001.txt</code>
<i>HP-UX 10.10 Patch List</i>	65	ASCII	<code>sw_patches.1010.txt</code>
<i>NFS Diskless Concepts and Administration</i>	514	Postscript	<code>NFS_Concepts_Admin.ps</code>

4-6 Pre-Upgrade Tasks for All 9.x Systems

To load the tools and online documents from the “Upgrade Tools” package onto a 9.x system, do the following:

1. Put the “Upgrade Tools” tape or CDROM into the drive.
2. Log into your 9.x system as `root`.
3. Run `/etc/update`.
4. Choose `Select All Filesets on the Source Media`.
5. After all of the files are loaded, remove the tape or CDROM from the drive and store it in a safe place.

To retrieve the online documents from tape before you load the tools, do the following:

1. Write-protect the tape according to its documentation.
2. Insert the tape into the drive.
3. `cd` to the root directory.
4. Load the documents onto your system:

```
tar -xvf /dev/rmt/0m UPG-DOC
```

or run `/etc/update` and load the UPG-DOC fileset.

5. `cd` to the `/etc/newconfig/10RelNotes` directory, which contains the release notes and white paper files.
6. To print the documents, use the `lp` command (and the `-ops` option if you are printing `filesys.ps` or `NFSD_Concepts_Admin.ps`).

To load the tools from a CDROM onto a 9.x system: Use `/etc/update` to load the UPG-DOC fileset.

Understanding the Changes

By now you should have the pre-upgrade tools and documentation on your system (see “Locating and Loading Tools and Documentation”, earlier in this chapter). You do not need to read all of the documentation before you begin to do the tasks in this document; this section indicates the minimum you need to read, with estimates of how long it is likely to take you.

Before You Run Pre-Upgrade Tools

Before going on, everyone must read:

- From *Release Notes for HP-UX 10.10*:
 - The following subsections under “Major Changes for HP-UX 10.0 and 10.01”:
 - “New Features of HP-UX 10.0 and 10.01 ”
 - “The HP-UX 10.0 File System”
 - “I/O Convergence”
 - “Software Distributor”
- The *HP-UX 10.0 File System Layout* White Paper
- The *NFS Diskless Concepts and Administration* White Paper, if you plan to create an NFS Diskless cluster.

This should take you two to six hours, depending on how fast you read and whether or not you already have some knowledge of the topics (from other vendors’ systems, or from an HP class, for example).

Before You Upgrade to 10.01

Before upgrading to 10.01 (but not necessarily before doing the pre-upgrade tasks in this chapter) you should also read:

- Chapter 8 in this manual.
- “Kernel Convergence” in the chapter on “Major Changes for HP-UX 10.0 and 10.01” in the *Release Notes for HP-UX 10.10*.
- “Other Operating System and Subsystem Changes” in the *Release Notes for HP-UX 10.10*.

This should take less than three hours to read.

If the 10.01 system will continue to operate with 9.x systems, you should also read the *HP-UX 9.x/10.x Interoperability Guide* (HP part number 5963-8920). A printed copy of that manual is included in the “Upgrade Tools” package.

Cluster

If you are going to upgrade an HP-UX (“DUX”) cluster, you should also read:

- The section on “NFS Diskless” in the *Release Notes for HP-UX 10.10*.
 - The *NFS Diskless Concepts and Administration* white paper, particularly the section on setting “shared” and “private” policies (you’ll find a similar discussion under “Planning Your Cluster Policies” in chapter 11 of the 10.01 version of the *HP-UX System Administration Tasks* manual) and, for important information on disk space and swap, the “Questions and Answers” near the end of the document (included in slightly abridged form as Appendix B of this manual).
-

Changing Existing Procedures

This section suggests procedures you should implement *while still running 9.x*, in order to make system administration easier once you are running 10.01.

Backups

As of 10.0, the structure of HP-UX has changed: when you come up on 10.01 or 10.10 many system files and commands will be in new locations, and in some cases these files and commands will also cause things to happen differently from the way they did on 9.x.

This means not only that you usually cannot recover 9.x system files onto a 10.x system and expect them to work, but that you should also guard against such a recovery happening accidentally—because system (or “structural”) files were backed up on the same tape as user files, for example.

“System” or “structural” files are those that form part of HP-UX, in particular files in / (**root**), /**bin**, /**etc**/, /**usr/bin** and /**usr/lib** on a 9.x system.

Note

- When you upgrade to 10.01, the upgrade software will preserve as much of your customization as possible in the “structural” files.

If you then decide to upgrade to 10.10, your customization will be preserved as in any other HP-UX upgrade.

- You may still have to modify some 10.01 system scripts, for example to preserve code you added to 9.x startup and shutdown scripts.

See Chapter 7 for information on modifying system startup and shutdown files, and Chapter 8 for guidelines on recovering (and moving and mounting) files from 9.x to 10.x.

Caution

HP supports only one method of upgrading from 9.x to 10.x: that is, running the 10.01 upgrade software to get to 10.01, then upgrading to 10.10 if you so decide. (9.07 systems *must* upgrade to 10.10, via a special version of 10.01 that is supported only for that purpose.)

If you *install* 10.x onto a system that was previously running an earlier HP-UX release, you will have to re-customize the system from scratch. If you try to merge customized 9.x system files with their 10.x counterparts, you are unlikely to succeed in producing a fully operational system, and HP will not be able to support the result.

Recommendation for Routine Backups on Existing 9.x Systems

HP recommends that you implement the following policies on your 9.x systems as soon as possible.

1. Begin backing up “data” files (user and application directories) separately from “structural” files (HP system directories), so that a given tape (or archive volume) contains only “data” *or* “structure”—only user files or system files.

For the purposes of this discussion, “system” files are HP-supplied files in the following 9.x directories:

```
/usr (particularly /usr/bin and /usr/lib )  
/etc  
/bin  
/lib  
/dev  
/system
```

Most 9.x files in these directories will not work on 10.x. *You should never recover them to their original (9.x) pathnames on a 10.x system.*

2. Use **fbackup(1M)** to do backups, or back up files to relative rather than absolute pathnames.

See Chapter 8 for more information on recovering, moving or mounting files from 9.x systems onto a 10.x system.

3. If your system is an HP-UX cluster and it stores context-dependent files (CDFs) in directories other than the HP “system” directories listed above (for example, if you have created CDFs yourself, or if you have applications that reside in CDFs or that create CDFs at run-time) you also should begin backing up these directories separately.

As of 10.0, the “hidden directories” that comprise CDFs are treated as ordinary directories. For example, the CDF element `/myapp/outputfile+/node1` will become an ordinary file (`node1`) in the ordinary directory `/myapp1/outputfile+` when you use `tar` or `frecover` to restore it to a 10.x system.

See “Preparing for Changes to HP-UX Clusters”, later in this chapter.

Note

- Avoid recovering HP-created CDFs onto 10.x systems; these are “structural” files and will not work on 10.x (see the guidelines for “structural” files above).
 - If you mount a disk containing CDFs onto a 10.x system, you won’t see the plus sign in the directory name, but the sticky and directory bits will be set.
-

Before You Go On

Before you continue with the rest of the tasks in this chapter, make sure you have done the following:

1. Loaded tools and documentation onto your system.

See “Locating and Loading Tools and Documentation” earlier in this chapter.

2. Read essential documentation.

See “Understanding the Changes” earlier in this chapter.

You should also make sure that the following manpages are on your system:

```
snoop(update_aid)      fscanner(update_aid)
prepare(update_aid)    analysis_dbs(4)
analyzer(update_aid)  upgdiskless(1M)
fnlookup(update_aid)
```

Note

- Unless otherwise stated in the procedures that follow, you *do not* need to be in single-user mode to perform the tasks in this chapter.
- Running `snoop` will not change the configuration of your 9.x system in any way.

But bear in mind that:

- `snoop` may slow the system when it’s running, especially on a “DUX” cluster server.
- You may need to make configuration changes in light of what `snoop` tells you.

For example, you may have to add or free up disk space.

See “Analyzing Your System” later in this chapter.

Preparing for Changes to HP-UX Clusters

(If you will not be upgrading any HP-UX (“DUX”) clusters, you can skip this section.)

HP-UX 10.01 replaces HP’s proprietary mechanism for clusters, often called “DUX,” with NFS technology, NFS Diskless.

Most of the work of converting a Series 700 cluster to NFS Diskless will be done for you by the upgrade software, but there are some important tasks you need to do yourself, particularly if your Series 700 cluster supports Series 300 and 400 clients—these computers are not supported as of 10.0, so you will have to make other provisions for them before upgrading the cluster: see “Providing for Series 300 and 400 Clients” in Chapter 2 for some suggestions.

The following sections summarize what the pre-upgrade tools and the upgrade software do to convert the cluster, and what you need to do yourself. See also “Converting HP-UX (“DUX”) Clusters” in Chapter 2.

What the Tools Do

- The `snoop(update_aid)` tool checks your system for:
 - Series 300 and 400 clients.
 - “Invalid” clients:
 - Clients that have entries in `/etc/clusterconf` but lack kernel or critical configuration files.
 - Clients whose entries in `/etc/clusterconf` have been “commented out”.

All other clients are considered “valid”, although full conversion is possible only for clients that are actually booted to the server during the upgrade.
- The upgrade software:
 - Removes the Series 300/400 operating system if it finds one on the server.
 - Preserves all HP-UX “system” CDFs for the server and “valid” Series 700 clients as regular HP-UX files.
 - Upgrades the server to HP-UX 10.01.

Caution

- The NFS Diskless software you will need on 10.x will be automatically selected for you during the upgrade when you choose **Match What Target Has**. **DO NOT DESELECT IT!**

See “What Is Match-What-Target-Has?”, later in this chapter.
 - The upgrade software does *not* follow symbolic links when searching for and converting CDFs.
-

- Post-upgrade tools:
 - Reclaim and reassign swap space on the server.
 - Add back clients as NFS Diskless clients.
 - Reconfigure networking, peripherals, etc. on the clients, using the 9.x configuration files and your input.

What You Need to Do (Summary)

Use this list as a checklist; details are later in this chapter and the chapters that follow.

■ Before you upgrade:

1. Run `snoop(update_aid)`.
See “Analyzing Your System”, later in this chapter.
2. Make other provisions for Series 300/400 clients.
See “Providing for Series 300 and 400 Clients” in Chapter 2.
3. Remove any Series 300 or 400 clients and any client “fragments” from the cluster.
See “Removing Series 300/400 Clients and Inactive Clients”, later in this chapter.
4. Re-run `snoop` to check that the cluster is ready to be upgraded.
5. Copy any non-“system” CDFs to regular files; back up those files; remove the non-HP CDFs.
See “Handling Non-“System” CDFs”, later in this chapter.
6. If the cluster is a Trusted System, “unsecure” the system so it can be upgraded.
Do this just before you run the upgrade. See ““Unsecuring” a Trusted System”, later in this chapter.

- After upgrading to 10.01:
 1. Run tools to convert and add back clients.
See “Upgrading HP-UX (“DUX”) Cluster Clients” in Chapter 6.
 2. Boot the clients.
 3. Fine-tune client configuration as needed.
 4. Upgrade applications, add optional HP-UX features, and upgrade the cluster to 10.10 if you so decide.
 5. “Resecure” a Trusted System cluster.
See Chapter 7.

Preparing To Upgrade a 9.07 System

9.07 systems pose a problem because they include graphics and other capabilities that are not supported on 10.0 or 10.01. This means that you *must* upgrade all the way to 10.10 from 9.07. But, to allow the file-system and other 9.x-10.x conversions to occur, 9.07 systems must still go *through* 10.01 on their way to 10.10.

What this means is that you will upgrade your 9.07 system, like any other 9.x system, initially to 10.01, but you don't have the option, as you do on other systems, of *staying* on 10.01: immediately after the upgrade you must load any applications you are upgrading from 9.x, and then update the system to 10.10. You will see warning messages about this as soon as you log in after the 10.01 upgrade completes.

In addition, you must give the upgrade software explicit permission to go ahead with the upgrade. You do this by running `snoop` and answering “yes” to the question:

```
Do you want to upgrade from 9.07 to 10.10?
```

See “Running `snoop`”, later in this chapter.

Note If you have many, similarly configured 9.07 systems, you do not have to run `snoop` interactively and answer this question on each one. You can bypass the question in `snoop` by exporting the environment variable

```
ALLOW_907_UPGRADE=yes
```

This allows both `snoop` and `upgrade` to run unattended, and commits you, once the upgrade is in progress, to upgrading the system all the way to 10.10.

Analyzing Your System

What snoop Does

`snoop` analyzes a Series 700 or Series 800 9.x system and reports conditions that may prevent you from successfully upgrading the system to 10.01. For example, you may need more disk space, or you may be trying to upgrade an HP-UX (“DUX”) cluster server that still has Series 300 or 400 clients (see “Converting HP-UX (“DUX”) Clusters” in Chapter 2).

Note

Even though your goal may be to upgrade to 10.10, you must upgrade to 10.01 first. And before you can upgrade to 10.01, you must do the preparation steps described in this chapter.

This is true even for 9.07 systems, which *must* upgrade to 10.10 as soon as they have loaded the 10.01 operating system and applications (see “Preparing To Upgrade a 9.07 System”, earlier in this chapter).

`snoop` consists of two parts. The first part analyzes the system configuration, checks for any problems that might prevent you from upgrading this particular configuration to 10.01, and gives you the opportunity to make changes in the configuration files that will be moved into place when you upgrade to 10.01.

The second part runs the Software Distributor tool `swinstall` *in preview mode*—your 9.x system is *not* affected. This allows for disk space and other checks to make sure your system is ready to be upgraded. The sections that follow provide detail about some of the new things you will see during the `swinstall` preview.

See the `snoop(update_aid)` manpage for a detailed description of `snoop`; see “Running `snoop`”, later in this chapter, for directions for running `snoop`.

What You Will See

You will be seeing a preview of the actual upgrade, including **Software Distributor** (SD) the new tool that replaces `/etc/update` on 10.x systems. This will all look unfamiliar, but it is easy to use, and running `snoop` will give you a chance to get used to SD's screens before you do the actual upgrade.

As well as getting used to SD itself, you also need to understand some new terms, in particular **bundles**, **products**, and **Match What Target Has**.

Briefly, a new, factory-installed HP-UX 10.01 system is built from software **bundles**, whereas existing 9.x systems are built from filesets. 9.x systems must be upgraded via a process called **Match What Target Has**. You can supplement this process if necessary by adding new 10.01 features packaged in **products** or **bundles**.

The following sections provide further explanation.

What Is Match-What-Target-Has?

When you run `snoop`, you'll see a message advising you to choose **Match What Target Has** (see "Running `snoop`", later in this chapter). "Target" is the system you are going to upgrade (as opposed to a "source" system that might store the software for a network upgrade).

This means that, when you do the actual upgrade, you should *initially* upgrade to a 10.01 system that is functionally equivalent to your 9.x system; that is, it will have the HP-UX 10.0 file system layout and the new versions of HP-UX commands and libraries, and it will have new features that replace 9.x features that are not supported on 10.x: for example, systems with SDS disks will have LVM; a cluster server will have NFS Diskless.

Caution	Always choose Match What Target Has and DO NOT DESELECT software automatically selected for you by Match What Target Has .
----------------	--

Choosing **Match What Target Has** ensures that your 9.x system upgrades successfully to 10.01, but this system will not have all of the new features that became available as of 10.0 and 10.01, such as the **Journalled File System** (JFS), and it will not have 10.10 features.

4-20 Pre-Upgrade Tasks for All 9.x Systems

You may or may not want these new features.

- If you don't want any of the additional 10.x features, then all you need to do is choose **Match What Target Has**.

Do this both when you run **snoop**, as described later in this chapter under “Running snoop”, and when you do the actual upgrade, as described in the next chapter.

In this case, you can skip the subsections that follow and proceed to “Running snoop”.

- If you want some of the additional 10.x features, or all of them, you need to do things a little differently when running **snoop** from when you do the actual upgrade.

“Adding Software after Match-What-Target-Has”, later in this chapter, explains what to do. But first you need to understand the difference between **products** and **bundles**.

What Are Bundles and Products?

If you have installed previous releases of HP-UX, you will be accustomed to seeing and choosing the software to be loaded in terms of **partitions** and **filesets**. SD also uses filesets, but introduces two new categories, **products** and **bundles**.

Products are similar to 9.x **partitions**; they are groupings of filesets.

Bundles are also groupings of filesets, but they are often larger than **products**, including filesets from two or more products. Their purpose is to group together functions that logically belong together; in fact some bundles, such as the **Runtime Bundles**, comprise a complete, standard HP-UX system.

When you run **snoop**, and later **upgrade**, SD will tell you to choose **Match What Target Has**. You must choose this option to be sure of getting a 10.01 system that works correctly; see the previous section, “What Is Match-What-Target-Has?”

But you may also want to add some of the new features added to HP-UX at 10.0 and 10.01, such as JFS (or VxFS, the Journaled File System) or, on Series 700s, LVM (Logical Volume Manager).

To get these new features, or improvements such as expanded localization support (NLS), you still need to upgrade to 10.01 via **Match What Target Has**, but then after the upgrade completes you will go back and add new software by choosing either specific **products** or one or more **bundles**.

Note You should add this new software immediately after the upgrade to 10.01, even if you are upgrading to 10.01 only in order to get to 10.10; this allows 9.x-to-10.x configuration and conversion work to take place.

Bundles or Products? As the procedure later in this chapter explains, SD allows you to look at and choose software at the bundle or the product level. These are called “views” in SD (and, as in `/etc/update`, you can also go down to the fileset level).

Each of these “views” of the software has advantages and disadvantages:

4-22 Pre-Upgrade Tasks for All 9.x Systems

- The advantage of choosing **products** is that it sometimes gives you greater control: you are choosing software at a lower level of detail.
 - The disadvantage is that you may not load everything you actually need.

Or, conversely, you may load more than you need; for example, loading X11 via the bundles view will install about 55 Mb; via the product view it is close to 90 Mb.
- The advantage of choosing a **bundle** (or **bundles**) is that these are designed to incorporate all the functionality you need.

For example, the **English Runtime Bundle** contains a complete HP-UX 10.01 English-language system, including new features such as LVM and JFS, and excluding filesets for foreign languages that English-speaking users are unlikely to need.

Note

- If you are upgrading a system that runs in, or uses, a European language other than English, make sure you have a copy of the *Localized System Configuration Overview* (HP part number B3782-90064).

Read the chapter in your language(s) before you run **snoop**.
- If you are upgrading a system that runs in, or uses, an Asian language, make sure you read the section on “ASE Delivery Restructuring” under “Bundles” in chapter 4, “Other Operating System and Subsystem Changes”, of the *Release Notes for HP-UX 10.10*.

Read the localization *Release Notes* or other local documentation in your language(s).
- If you operate small Series 800 systems that do not use X-windows and related services, you should pay particular attention to the “non-graphics” bundles discussed under “Non-Graphics Bundles” later in this chapter.

□ The disadvantage of bundles is that you may load more than you need.

- This is not a problem if you have plenty of disk space.

You can always remove the filesets you don't need later. See "Removing Software You Do Not Need" in Chapter 7.

- It is a problem if you are short of disk space.

Make sure that when you run `snoop`, you choose not only **Match What Target Has**, but also any additional bundles (or products, or both) you intend to load. `snoop` will tell you if you are choosing more than will fit on your system, and you will be able to examine the contents of the bundle to see if there parts of it you don't need. See "When To Use Products", later in this section, for more information.

When To Use Bundles. It is usually better to add a bundle or bundles, rather than products, to the 10.01 system built by **Match What Target Has**.

If you want all of the additional features, choose a **Runtime Bundle**; they include everything for a system running in a given human language.

From now on, this manual assumes that you will add new features to your 10.01 system by choosing bundles, rather than products; but this does not mean you *must* choose bundles. The bullets that follow indicate cases in which it may be better to work with products.

When To Use Products.

- If you tell `snoop` you are going to load a bundle or bundles in addition to the software chosen by **Match What Target Has**, and `snoop` warns you you are short of disk space, then you may want to "unmark" some products in the bundle (meaning that you intend not to load them when you do the actual upgrade).

Don't "unmark" the bundle, but go down to the "products" level (double-click on the bundle name) and "unmark" any products you are sure you don't want; then try the disk space analysis again.

- If you are interested only in one particular feature that is packaged as a product (such as LVM), then add that product rather than a bundle.

4-24 Pre-Upgrade Tasks for All 9.x Systems

Adding Software after Match-What-Target-Has

If you know you want new 10.01 features such as LVM on the Series 700, or JFS (the Journaled File System, also known as VxFs) you will need to add software to the software chosen for you by **Match What Target Has**. This section explains how and when to do this.

How To Add and Preview Software.

■ **When upgrading (running upgrade):**

HP recommends that you proceed as follows:

1. Build your 10.01 system by running **upgrade** and choosing **Match What Target Has**.
2. After the upgrade completes, go back and load additional software if you need it.
3. Upgrade to 10.10 if you so decide.

If you are not sure whether or not you need a new product or feature that was not chosen by **Match What Target Has**, don't load it; you can always go back and add it later.

For more information, see "Running the Upgrade" in Chapter 6 and "Loading New 10.01 Features and HP Applications" in Chapter 7.

■ **When previewing (running snoop):**

As a preparation for loading additional software onto the 10.01 system (after the initial upgrade), you need to run **snoop** now, on the 9.x system, and choose this same software in preview mode.

For example, if you intend to add Logical Volume Manager to a Series 700 system, run **snoop** and:

1. Choose **Match What Target Has**.
2. Select LVM and choose **Mark for Install**.

Do this even if you intend to upgrade the system all the way to 10.10, and will never actually use LVM until you get to 10.10.

This allows **snoop** to do an accurate disk space analysis and warn you if you will not have enough space or are likely to encounter any other problems.

snoop's analysis is only as good as the information you give it about what you intend to load, so it's important to take the time to decide exactly what you need.

- See Appendix A for a list of 10.01 bundles and products; you can find out more about the contents of these when you run **snoop**; see “Running snoop”, later in this chapter.
- If you have not already done so, read “I/O Convergence” and “Other Operating System and Subsystem Changes” in the *Release Notes for HP-UX 10.10*.

See “Locating and Loading Tools and Documentation” for information on getting the *Release Notes for HP-UX 10.10* and related documentation from the “Upgrade Tools” tape or CD.

Summary.

1. When you run **snoop**:
 - Choose **Match What Target Has**, and also choose **(Mark for Install)** all the new features you intend to add.
2. When you run **upgrade**:
 - Choose *only* **Match What Target Has**, then let the upgrade complete.
3. After upgrading to 10.01, and before updating to 10.10:
 - Load new 10.01 features and upgrade 9.x applications to 10.01.

Why Two Steps for Upgrade? The reason HP recommends loading additional software *after*, rather than *during*, the upgrade, is that all of the work of creating a functional 10.01 system is done automatically when you run **upgrade** and choose **Match What Target Has**.

If you interfere with this process by adding to and subtracting from (**marking** and **unmarking**) the software that is to be loaded, you run the risk of not getting a complete, integrated 10.01 system. At worst, you could fail to load software that is needed to build a working kernel.

Running snoop

To analyze your system and preview what will happen when you load 10.01 software, follow the procedure that starts on the next page.

Log File

snoop writes a logfile `/var/adm/sw/snoop.log`.

- You can look at this log online while **snoop** is running.
- Successive runs of **snoop** will add to `/var/adm/sw/snoop.log` if it exists.

Cluster

If this system is an HP-UX (“DUX”) cluster, run **snoop** on the cluster *server*. You can be logged in to the server on the console or remotely from another system. Make sure all the clients you intend to upgrade are booted to the server.

Procedure

1. Log in as the root user.

Cluster

- Make sure you are logged in to the *server*.
- Check that all the clients are booted to the cluster:

```
cnodes -a
```

Clients listed with an asterisk are *not* booted to the cluster; **snoop** analysis will omit important information that can be obtained only from these clients.

snoop will copy and modify the `/.rhosts` file on the server (or create it if it doesn't exist) so that it can obtain information about each of the active clients via `remsh(1)`. Before the **snoop** program exits, it will put your original `/.rhosts` back in place.

-
2. If you are using a graphical display (for example a workstation monitor or Xterminal), set your `DISPLAY` variable, for example:

```
DISPLAY=hpulpcu2:0.0; export DISPLAY
```

This will allow the SD (Software Distributor) software, which **snoop** invokes, to run in graphical mode (with windows).

3. Put the 10.01 tape or CD in the drive (unless the 10.01 software is on a network server).

snoop needs this to estimate disk space needs; but this is a preview only; **snoop** will *not* load any software.

- If you are using a CD, make sure the CDROM device is mounted,

```
/etc/mount
```

If there is no entry for the CDROM device, mount it; for example:

```
/etc/mount /dev/dsk/2s0 /cdrom
```

The CDROM filesystem must be mounted to the local system; an NFS mount will not work. You can use a CDROM drive attached to a remote

system, but you must register the CDROM filesystem as a depot (see “Building an SD Depot”, later in this chapter).

- If you are using DDS tape, the drive must be attached to the system which is to be upgraded.

(If it is not, you’ll get an error message referring to a “single marked target”. This means “the local system”. You may want to look into building a depot on disk; see “Building an SD Depot”, later in this chapter.)

4. Run `snoop`:

```
/usr/sbin/snoop
```

This starts `snoop` in its default interactive mode.

5. Respond to the questions that `snoop` asks you:

9.07: `snoop` will warn you that you must upgrade all the way to HP-UX 10.10, and ask you to confirm that you want to do this; respond yes unless you really don’t want to upgrade this system.

- If you have run `snoop` before, it asks you if want to overwrite the configuration files created on the previous run.

You probably don’t want to do this unless you have changed your mind about some aspect of the configuration; you certainly don’t want to do it if you have already modified these files according to `snoop`’s directions (see next page).

- `snoop` lists environment variables and system utilities and tells you how they will be set when 10.01 comes up on your system. (The settings correspond to the 9.x settings where applicable.)

- If you want to change any of these values, respond **y** when **snoop** asks you if you want to modify them.

snoop will then take you through the list item by item.

Note These values are being saved for the 10.01 system; your 9.x configuration will not be changed in any way, whether or not you change the values in this list.

- If you're happy with the defaults, respond **n**.

If in doubt, respond **n**; you can always make changes once the 10.01 system is up and running.

The files **snoop** creates are saved in `/etc/upgrade/save/hostname/tttools`, where *hostname* is the official hostname of your system (as returned by `uname -n`). These files will be used to create 10.01 system configuration files when you do the upgrade to 10.01.

Cluster This is true for the server and each of the clients.

If you have customized any of your 9.x system configuration files, you may have used pathnames or commands that will not work on 10.01; **snoop** will warn you about this and direct you to any files that need additional modifications. Follow the directions in the log file (`/var/adm/sw/snoop.log`).

- **snoop** asks you if you want to change the boot LIF configuration.

“No” (**n**) is a safe answer to this question.

If you want the 10.01 system to boot from a different disk from the current (9.x) system, answer **y**. This will not affect the way the 9.x system boots.

- **snoop** asks you if you want to add to the boot LIF configuration.

Answer **y** if you have additional LIF volumes (i.e. more than one bootable device) that will need to be modified for 10.01.

4-30 Pre-Upgrade Tasks for All 9.x Systems

6. `snoop` now checks for conditions that will pose problems for the upgrade.

Caution

You *will not* be able to upgrade this system as it stands if `snoop` reports that:

- You have less than 16 Mb of memory.

You will need to upgrade to 16Mb. This is true for a standalone system, a cluster server, and all cluster clients.

- You have Software Disk Striping (SDS) on the root disk (Series 700).

HP has never supported this configuration.

See “Handling `snoop` PROBLEMS”, later in this chapter, for more information.

Cluster:

- You will not be able to upgrade an HP-UX (“DUX”) cluster server that has Series 300 or 400 clients.

You must remove these clients before you can upgrade; see “Providing for Series 300 and 400 Clients” in Chapter 2, and “Removing Series 300/400 Clients and Inactive Clients”, later in this chapter.

- `snoop` will also warn you about potential disk space problems, and changes affecting where and how the clients will swap. See “Reclaiming Device Swap”.

7. `snoop` is now about to invoke SD (Software Distributor) in preview mode, and asks you if you can use the Graphical User Interface.

Respond `y` if you’re using a graphical display. If you did not set the `DISPLAY` variable as recommended in step 2 of this procedure, `snoop` will prompt you.

Note This is *not* the actual upgrade; your 9.x configuration will not be changed.

8. The **Specify Source** window comes up.

Troubleshooting

Problem	What to do
(errors indicating X connection was refused)	If you are running snoop from a window on a remote machine, make sure that you have permission to connect to the system being upgraded: try entering <code>/usr/bin/X11/xhost + or /usr/bin/X11/xhost + hostname</code> , where <i>hostname</i> is the official hostname of the machine being upgraded.
FATAL ERROR: X Toolkit Error: Cannot perform malloc	Your system is short of memory or swap. Try closing windows, running snoop from a remote system, or both.

SD first looks for the 10.01 software under `/var/spool/sw` on your system's root disk; this is the default location for an SD software "depot". ("Depot" is SD's name for the software source from which you are going to upgrade; see "Building an SD Depot", later in this chapter, for information on building a software depot.)

There will not be any such directory unless you have built a "depot"; so SD defaults to `/dev/rmt/0m` instead. You can accept this default *unless*:

- You have loaded the tape into a drive addressed by a device file other than `/dev/rmt/0m`.

Or

- You are upgrading from a network server.

Enter the official hostname and the software source "depot", for example:

4-32 Pre-Upgrade Tasks for All 9.x Systems

```
Source:                upgserver
Source depot path:    /release/s700_10.01
```

You can also select from available servers and “depots” by clicking on **Source** and **Source depot path** respectively.

Caution **Cluster:** For an HP-UX (“DUX”) cluster, the depot must contain only the HP-UX core software, not HP Application Release (“AR”) software.

Or

- You are upgrading from CDROM.

Enter the name of the filesystem to which the CDROM drive is mounted (the **mount point**) for example,

```
/cdrom
```

9. The **Software Selection** window comes up.

Pull down the **Actions** menu and choose **Match What Target Has**.

This tells **snoop** that you intend, when you do the actual upgrade, to load a system that has functionality at least equivalent to your current, 9.x system.

HP recommends this option for all 9.x-to-10.01 upgrades. See “What Is Match-What-Target-Has?” earlier in this chapter.

You may also want some or all of the new features being offered in HP-UX 10.0 and 10.01; see “What Are Bundles and Products?” and “Adding Software after Match-What-Target-Has”, earlier in this chapter, for more information.

To tell **snoop** you will be adding software to the basic software chosen by **Match What Target Has**, do the following:

- a. Choose **Match What Target Has**.
- b. Pull down the **View** menu and and choose **Change Software View**.
- c. Choose **All Bundles**.

See the **Note** that follows for an explanation of the available “software views”.

- d. Select the “bundle” you need from the list now showing on your screen.

For example, for a minimum Series 700 system you might choose the “Desktop” bundle; for a larger 700, choose the “Runtime” bundle for the human language your system will use (for example, in an English-speaking country you would probably choose the **English Runtime** bundle).

For a small Series 800 system, you may want to choose a “non-graphics” bundle; see “Removing Unneeded Filesets” under “Handling snoop PROBLEMS”, later in this chapter, for a discussion of this option.

- e. Pull down the **Actions** menu again and choose **Mark for Install**.

Caution

When running **snoop**, you can safely add software to what SD automatically selects when you choose **Match What Target Has**; but *do not* do this when you run the actual upgrade.

See “Adding Software after Match-What-Target-Has”, earlier in this chapter, for discussion; for specific directions for running the upgrade and adding software, see “Running the Upgrade” in Chapter 6 and “Loading New 10.01 Features and HP Applications” in Chapter 7 respectively.

Note

The **Software Selection** screen allows you to look at the software selections in several different ways:

- **Products**
- **All bundles**
- **Bundles of one category**

Pull down the **View** menu to see these choices. **Products** will look familiar if you are used to upgrading earlier releases of HP-UX: it is the closest to the **partitions** in which filesets were grouped in earlier releases. **Bundles** are also groupings of filesets, but they may include filesets from more than one product.

You can find out more about a given selection by double-clicking on it; for example, double-clicking on a product will show you the filesets it contains; double-clicking on a fileset shows the files in it. Click on **Description** in the **Software Description Dialog** box to find out more about the item in question.

See “What Are Bundles and Products?”, earlier in this chapter, for more information.

10. The **Software Selection** window re-appears.

Pull down the **Actions** menu again, and this time choose **Install (analysis)**.

This brings up the **Install Analysis** window, which gives you the option of reading the analysis logfile as it's being written: click on **Logfile** to see the logfile. (This file is saved as `/var/adm/sw/swagent.log`.)

Troubleshooting

Problem	What to do
Cannot fork to run <i>filename</i>	Your system is short of memory or swap. Try closing windows, running snoop from a remote system, or both.
(snoop reports products or filesets not found.)	If the 10.01 software is in an SD “depot”, check that you have built the depot correctly and loaded everything you need from the tapes or CDs supplied by HP. See “Building an SD Depot”.

11. When the analysis is complete, you'll see a message such as **Ready** or **Ready with warnings**.
 - If there are warnings, check the logfile for details (click on **Logfile**).
 - Click on **Disk Space** to see if you have sufficient free disk space to load 10.01.
 - To get out of the program, click on **OK**.

Modifying Your System

After running `snoop`, do the following:

- Read the logfile `/var/adm/sw/snoop.log` paying particular attention to PROBLEMS (which you *must* be correct) and CAUTIONS (which you *should* be acted on or corrected). Also check `/var/adm/sw/swagent.log`.
- Perform the steps that are recommended in the logfiles.
- Read “Preparing for I/O System Changes” later in this chapter, and the sections that follow, and perform the tasks that apply to your particular system.
- Run `snoop` again until all PROBLEMS and as many CAUTIONS as possible are corrected.

This may mean running `snoop` a number of times. You don’t need to run `snoop` interactively each time. To run `snoop` unattended, enter a command such as:

```
/usr/sbin/snoop -a -o -s petunia:/mydepot
```

(This example uses a network server named `petunia` which has a 10.01 software depot under the directory `/mydepot`.)

Note

- If all of the problems reported are in the `snoop.log` logfile, you can re-run just `snoop` (without the SD preview) by using `snoop`’s `-d` option, for example,

```
/usr/sbin/snoop -a -d
```

Running `snoop` this way will *not* help you resolve problems reported in `/var/adm/sw/swagent.log`, and will not re-check disk space, so it is important to run `snoop` once more with `-a -o -s ...` (as described above) before you do the actual upgrade.

Each time you run `snoop`, new information is appended to the `snoop.log` logfile. This happens whether you run the program interactively or unattended.

If you want a fresh log each time you run `snoop`, you can simply delete the file `/var/adm/snoop.log` before re-running `snoop`.

Caution *Do not* remove the `/var/adm/sw` directory; this will remove files that the upgrade and pre-upgrade software depends on.

If you do accidentally remove `/var/adm/sw`, run `/etc/update` to re-install the SD- filesets shown in “Locating and Loading Tools and Documentation” earlier in this chapter.

See “Handling `snoop` PROBLEMS”, later in this chapter, for suggestions for solving PROBLEMS that `snoop` may report.

Modifying HP-UX Clusters

The subsections that follow provide instructions for modifying HP-UX (“DUX”) clusters so that they will convert smoothly to NFS Diskless during the upgrade to 10.01.

Removing Series 300/400 Clients and Inactive Clients

Before you get to this section, you should have made provision for your Series 300 and 400 clients, and they should be up and running in the new configuration. See “Providing for Series 300 and 400 Clients” in Chapter 2.

Caution DO NOT PROCEED with this section if Series 300/400 clients are still booted from the Series 700 server.

Now you need to clean up the Series 700 cluster. `snoop` has identified the Series 300 and 400 clients you need to remove (`snoop` refers to them as “68K” clients because of their 680n0 processor chips).

Do the following:

1. Remove Series 300 and 400 clients.

Use the SAM **Remove Cluster Clients** screen. Accept the option to remove all client-specific files as well as the clients themselves.

Caution

You *must* remove Series 300 and 400 clients before you begin upgrading to 10.01. The upgrade software will not do this for you; if it finds Series 300/400 clients, it will exit, leaving the clients intact.

2. Remove any inactive clients—clients you never intend to boot again from this server.

Use SAM and remove all the files as with the Series 300 and 400 clients.

You must also decide what to do about clients **snoop** has identified as having **PROBLEMS**. These will be clients that appear in `/etc/clusterconf` (in entries that have not been “commented out”) but do not have a kernel or critical configuration files. **snoop** refers to these as “invalid” clients. (Clients whose entries have been “commented out” in `/etc/clusterconf` are also reported as “invalid”.)

- If you intend to boot these “invalid” clients under NFS Diskless on 10.x, fix them as **snoop** indicates.

For example, build a kernel if that is what is missing; see the Series 700 9.0 manual *Managing Clusters of HP 9000 Computers* for more information.

- If you don’t intend to boot them from the 10.x cluster, use SAM to remove them now.

What if You Do Not Clean Up. You may be wondering about the consequences of not doing the work described in the previous subsection. They are as follows:

- If you do not remove Series 300 and 400 clients:

- The upgrade will fail, leaving the 9.x system intact. You'll see a message such as:

```
Cluster has either S300/400(68K) clients or other
unknown architecture clients. The UPGRADE process
does not support heterogeneous upgrade to 10.x.
User must remove 68K clients and then restart the
upgrade process.
```

- If you do not remove inactive clients:

- If the client does not have a kernel, all its critical configuration files, and a valid IP address, the conversion will not succeed, and you will see **WARNINGS** during the upgrade.

Depending on what parts of the “DUX” configuration are missing, some or all of configuration files that do exist may remain on the server after the upgrade.

For example:

- For a client lacking a valid IP address, but otherwise intact, the kernel and all HP-UX configuration files will be saved as regular HP-UX files (that is, not CDFs) on the server under `/etc/upgrade/save/client`;
- For a client “commented out” in `/etc/clusterconf`, or lacking a kernel or critical configuration files (“invalid client”) no files will be saved. (A kernel that is a symbolic link to a valid kernel file does not make the client “invalid”.)
- If the client has everything it needs to function under “DUX”, then you won't see any **WARNINGS** about missing files during the upgrade.

You will see messages indicating that the client is not booted to the cluster (assuming that it isn't; `snoop` and `upgrade` refer to such clients as “offline clients”); and the client will show up on the menus of the `upgdiskless` tool that you will run immediately after the upgrade completes on the

4-40 Pre-Upgrade Tasks for All 9.x Systems

server (`upgdiskless` completes the upgrade and conversion of the clients; see “Upgrading HP-UX (“DUX”) Cluster Clients” in Chapter 6).

Assuming you do not choose to activate the client via `upgdiskless`, its files will remain on the server under `/etc/upgrade/save/client`.

(If you do choose to activate a “valid” client that was not booted to the server during the upgrade, `upgdiskless` will build a minimally-configured client, using defaults similar to those for a new client created in SAM. See “If Clients Are Not Booted” in Chapter 2 for details.)

Summary: As you can see, failure to remove Series 300 and 400 clients is a showstopper. The other cases are less serious; the main consequence is that you consume disk space unnecessarily—depending on how much free disk space you have, this may or may not be a serious problem.

Handling Non-“System” CDFs

CDFs are context-dependent files, the files that are “owned” by particular nodes within the global file-system. See chapter 2 of the 9.0 version of *Managing Clusters of HP 9000 Computers* if you need more explanation.

Some clusters contain only HP-created “system” CDFs (kernels, device files and system configuration files such as `/etc/inittab`). If this is true of your system, you can skip the next subsection: the upgrade software will convert HP “system” CDFs to regular files and make them available to the nodes that need them on the 10.01 system.

If you have created CDFs, or if you run applications that create CDFs or are themselves stored as CDFs on the system, then you have another task to do.

Saving Non-“System” CDFs. Non-HP CDFs, and CDFs used by HP applications that are not part of the core system, will appear to the 10.x file system as ordinary directories. This probably is not what you want.

Note If the CDFs are in the 9.x “system” directories `/bin`, `/etc`, `lib`, `/system`, `/tmp`, or `/usr`, they will be saved on the server as regular (non-CDF) files, as `/directory/CDFname` for the server’s element and as `/directory!/CDFname/element` for the other elements.

If you know where they reside in the file system, you can prepare non-“system” CDFs for the upgrade as follows:

1. Copy each element to a regular HP-UX file.

Make the names relate in some obvious way to the nodes that own the files. For example, you might copy `/apps/myapp+/node1` to something like `/10.copies/myapp.node1`.

2. Make a backup of the regular files.

Delete the copies on disk if you like.

3. If you have any *directories* that are CDFs (apart from those created by HP-UX when you configured the cluster) repeat the above steps for the directories.

Copy the elements to a regular directory and back the regular directories up to tape.

4. *Just before upgrading to 10.01, and after you have done a backup with the -H option*, remove all non-“system” CDFs from the system.

The `-H` to backup commands such as `fbackup(1M)` backs up all elements of CDFs.

See “Backing Up Your System” at the end of this chapter.

Reclaiming Device Swap

If you see a **WARNING** in the `snoop` logfile that you have more device swap on the server than you are likely to need under NFSD, you should consider doing two things:

1. Create file systems on disks that are currently dedicated to device swap.
Plan to do this during the period when the cluster is down for the upgrade, either just before or just after you upgrade the server, but before trying to boot the clients under NFS Diskless.
2. Create a file which will tell `upgrade` to reclaim device swap at the end of file systems on the server's disk(s) and add it back to the file system.

See “Creating the Swap Reclamation File”, later in this section.

Note

The default for NFS Diskless clients is to swap to the server's file system space. You cannot configure *device* swap to any remote disk, including the server's. Make sure you understand how `upgrade` and `upgdiskless` (the program you will run right after the upgrade to re-activate the clients) convert various “DUX” swap configurations to NFS Diskless swap: see the subsection on “Swap” under “Upgrading a “DUX” Cluster” in Chapter 2.

Creating the Swap Reclamation File. Create the swap reclamation file as a plain ASCII file, using your favorite editor.

It must contain entries in the form:

```
character_device_filename Mb_to_reclaim
```

For example:

```
/dev/rdisk/c201d0s0      80
/dev/rdisk/c201d1s0      155
/dev/rdisk/c201d2s0      107
```

Note You must supply the *character* (or “raw”) device-file name, not the block device file name. By default, character device files are stored in the directory `/dev/rdisk`.

Entries must be one line per disk, fields delimited by one or more space characters (any combination of spaces and tabs).

Comment lines preceded by the pound character (`#`) are allowed.

Save the file in following path:

```
/etc/upgrade/save/server/ttools/AllSwap.d/reclaim_swap
```

where *server* is the name of the server as returned by `uname -a`. `snoop` has already created the directories in this path.

When you run `snoop` again, `snoop` will check for the file and check the validity of the entries. You will see **CAUTIONS** if there are errors in the file; if the file does not exist you'll see the same **CAUTION** as before about excessive device swap. If the file exists and has no errors, you'll see a **NOTE** that `snoop` found a valid reclamation file, and a list of its contents.

“Unsecuring” a Trusted System

If the cluster is a Trusted System, you will need to “unsecure” it before you can upgrade it. You will probably want to wait to do this until just before running the upgrade.

To “unsecure” the cluster, enter the following command on the server:

```
/etc/tsconvert -r
```

After the upgrade to 10.01 is complete, and you have run `upgdiskless` to add back the clients, you will need to “resecure” the cluster; see “Resecuring a Trusted System Cluster” in Chapter 7.

Handling snoop PROBLEMS

This section provides suggestions for correcting PROBLEMS you may find in the logfile `/var/adm/sw/snoop.log`.

If You Have Less than 16 Mb of Memory

You cannot upgrade a system that has less than 16Mb of memory. This includes cluster clients.

Either:

- Order more memory; or
- Do not upgrade this system.

If you decide not to upgrade the system, remember that you can still NFS mount file systems to and from other 10.x systems in the network, so long as you do not try to mount *system* directories. See “Moving Files” in Chapter 8, and “Operating 9.x and 10.x Systems Together” in Chapter 2, for more information.

If You Have an 8Mb Memory Card in Slot One (Series 800)

If you got a warning from `snoop` about this, it does not necessarily mean that you have this configuration, only that you *could* have.

To find out for sure, do the following:

1. Reboot the system and interrupt the boot by pressing any key.
2. Answer `n` when prompted to boot from primary path.
3. Answer `n` when prompted to boot from alternate path.
4. Answer `y` when asked if you want to interact with ISL.
5. At the ISL prompt, enter `iomap`.
6. You'll be asked if you want to modify program parameters; answer `n`.

Now you'll see a display at the end of which the memory cards are listed (under **Memory Controller**).

If the first card listed is an 8Mb card, you cannot upgrade this system as it stands.

- If you can swap another memory card into this slot, do so.
 - If this is not possible, or you are not comfortable doing it, contact your HP representative for help.
7. `iomap` asks you if you want to return to ISL. Answer `y`.
 8. Boot the system:

`hpux`

If There Are “PROBLEMS” on an HP-UX Cluster Server

- If `snoop` reports that you still have Series 300/400 clients, you *must* remove them before you can upgrade the cluster. See “Removing Series 300/400 Clients and Inactive Clients”, earlier in this chapter, and “Providing for Series 300 and 400 Clients” in Chapter 2 for details.
- If `snoop` reports that some of your clients were not booted to the server, you should, if possible, boot these clients to the server and re-run `snoop`.

This is the only way to be certain of catching all potential problems before you do the actual upgrade. If the clients are not booted to the server during the upgrade itself, they will, at best, lose much of the benefit of the conversion (see “If Clients Are Not Booted” in Chapter 2); at worst (if there were problems that have gone undetected because the clients were not booted during any run of `snoop`) the clients may not be converted at all.

If you never intend to boot these clients from this server under 10.x, you should remove them from the cluster before you do the upgrade; see “Removing Series 300/400 Clients and Inactive Clients”, earlier in this chapter.

- If `snoop` reports that it was unable to convert local swap defined in the `dfile` (the client’s kernel-configuration file) to a swap entry in `/etc/checklist`, you will need to create the entry in `/etc/checklist` yourself.

Modify `/etc/checklist+/client`, where *client* is the name of the client reported by `snoop`. The *HP-UX System Administration Tasks* manual explains how to create swap entries in `/etc/checklist`.

Note You also need to create a swap entry in `/etc/checklist+/client` if you have activated local swap only from the command line (with a `swapon` command)—that is, if there is no entry in the client’s `dfile` or `checklist` file to make the local swap “permanent”.

- If `snoop` reports that this is a Trusted System, you need to “unsecure” the system before the upgrade can proceed; see ““Unsecuring” a Trusted System” earlier in this chapter.

4-48 Pre-Upgrade Tasks for All 9.x Systems

If You Do Not Have Enough Disk Space

`snoop`'s disk space analysis errs on the conservative side; that is, it's possible `snoop` may tell you you don't have enough space when in fact you have just enough, but it is unwise to gamble unless this is a test or non-production system and you can afford the time to experiment.

Note An HP-UX ("DUX") cluster server is more likely to fail disk space analysis than other systems, because the algorithm is allowing for file-system swap to the server's disk space. See "Swap" in Chapter 2 for a discussion.

Remedies

Remedies depend on the nature of the problem `snoop` reports:

1. If you don't have as much total free space on your system as `snoop` says you need, you may have to add a disk or disks before you can upgrade.

But you may have more software on your system than you need, or you may have told `snoop` (by "marking" additional bundles or products) that you intend to add 10.01 features that you don't actually need.

- See "Removing Unneeded Filesets", later in this chapter, for suggestions on trimming down your 9.x system.
 - See "What Are Bundles and Products?" and "Adding Software after Match-What-Target-Has", earlier in this chapter, for information on choosing software in addition to what is chosen automatically by `Match What Target Has`.
2. If you have enough space in total, but not on the root disk, you may be able to redistribute the disk space.

- On a Series 800:

- If the problem is with / (**root**), do one of the following:

- If using LVM, move the root volume to a larger logical volume.

- This requires an unused disk that will become the new bootable device. Use the procedure that begins on page 8-121 of the 9.0 version of the Series 800 *HP-UX System Administration Tasks* manual.

- If using hard partitions, move directories under / (for example /**tmp**) to another partition.

Note

This will not work for directories such as /**etc**, /**lib** and /**dev**.

- If using hard partitions, move the root partition.

- See Chapter 6, “Managing the File System,” in the 9.0 version of the *HP-UX System Administration Tasks* manual for details.

- If the problem is with /**usr**, do one of the following:

- If using LVM, extend the /**usr** volume.

- See Chapter 8, “Managing Logical Volumes,” in the 9.0 version of the *HP-UX System Administration Tasks* manual for details. This procedure cannot be performed via SAM.

- If using hard partitions, move /**usr** to a larger partition.
 - Move /**usr** to another disk.

- See Chapter 6, “Managing the File System,” in the 9.0 version of the *HP-UX System Administration Tasks* manual for details.

- On a Series 700:

- Move /**usr** to another disk.

- See Chapter 6, “Managing the File System,” in the 9.0 version of the *HP-UX System Administration Tasks* manual for details.

4-50 Pre-Upgrade Tasks for All 9.x Systems

Removing Unneeded Filesets

When you choose **Match What Target Has** during the upgrade, SD will load all software that corresponds to what is on your 9.x system. If **snoop** reports you are short of disk space, make sure, before you do the upgrade, that you do not have filesets on your system that you don't need.

Likely candidates for removal are Native Language Support (NLS) and (on Series 800) graphics filesets. See “Series 800 Small System”, later in this chapter, for more information.

If you have the HP-UX “manpages” in hardcopy or on CDROM, you may want to remove the online version.

Use the **rmfn** utility, described in chapter 2 of the 9.0 *HP-UX System Administration Tasks* manual, to remove 9.x filesets you are not using and whose functionality you will not need on 10.x.

Note If you have more than one user licence on your system (for example, a 2-user licence *and* an 8-user licence), remove the licence you do not use before you upgrade. Check the **/system** and **/etc/filesets** directories for licence files such as **O2-USER** and **O8-USER**.

Series 800 Small System. By “small system” we mean a system that probably has these characteristics:

- 16-32 Mb of RAM.
- Single hard disk of 500Mb or less.
- Disk space is at a premium.
- System does not use graphics (X11, Vue, etc.) or Native Language Support (NLS).

The last two bullets in the above list are crucial; read the discussion below, and “Filesets To Remove”, later in this chapter, before you make a decision.

HP-UX 10.01 includes **bundles** (groupings of products and filesets) for the Series 800 that include the run-time system, but exclude NLS and graphics filesets that many Series 800s do not need. If you intend to install one of these bundles on your Series 800, do the following when you upgrade to 10.01:

- Choose Match What Target Has
- Proceed with the upgrade.
- Once the initial upgrade has succeeded, run SD again and this time select the “non-graphics” bundle for the human language that the system will use.

See the next subsection, “Non-Graphics Bundles”.

These bundles include features such as JFS that are new for 10.01.

- Select any other bundles you have ordered in addition to HP-UX.

See “Loading New 10.01 Features and HP Applications” in Chapter 7 for more information.

This can save you more than 50Mb of disk space, but *only if you first remove the graphics and NLS filesets* listed under “Filesets To Remove” below.

Non-Graphics Bundles. The following are the names of the non-graphics bundles as they appear on SD’s **Software Selection** screen when you set **software view** to **All bundles**. Choose the bundle that corresponds to the human language your system will use; for example, if the system’s users are primarily English-speaking, you would choose **HPUXEngRT800**.

Bundles	Descriptions
HPUXDanRT800	Danish non-Graphics Runtime HP-UX Environment
HPUXDutRT800	Dutch non-Graphics Runtime HP-UX Environment
HPUXEngRT800	English non-Graphics Runtime HP-UX Environment
HPUXFinRT800	Finnish non-Graphics Runtime HP-UX Environment
HPUXFreRT800	French non-Graphics Runtime HP-UX Environment
HPUXGerRT800	German non-Graphics Runtime HP-UX Environment
HPUXItaRT800	Italian non-Graphics Runtime HP-UX Environment
HPUXJpnRT800	Japanese non-Graphics Runtime
HPUXKorRT800	Korean non-Graphics Runtime
HPUXNorRT800	Norwegian non-Graphics Runtime HP-UX Environment
HPUXSchRT800	Simp.Chinese non-Graphics Runtime
HPUXSpaRT800	Spanish non-Graphics Runtime HP-UX Environment
HPUXSweRT800	Swedish non-Graphics Runtime HP-UX Environment
HPUXTChRT800	Trad. Chinese non-Graphics Runtime

4-52 Pre-Upgrade Tasks for All 9.x Systems

Filesets To Remove. The following lists show what filesets you can remove from your 9.x system if you do not use graphics, or Native Language Support, or both.

Graphics Filesets
=====

AUDIO
AUDIO-MAN
BMS
CAPTURE
IMAGING-RUN
IMAGING-SHLIBS
IMAGNG-RUN-MAN
SHPRNT-HELP
SHPRNT-PCL
SHPRNT-INSTALL
SHPRNT-PCONFIG
SHPRNT-CLI
SHPRNT-NLS
SHPRNT-SRV
X11-FONTA
X11-FONTB
X11-FONTC
X11-FONTSRV
X11-RUN
X11-RUN-HELP
X11-RUN-MAN
X11R4-SHLIBS
X11R5-SHLIBS

NLS Filesets
=====

Language -----	Fileset Name -----
Arabic	ARABIC ARABICW
Bulgarian	BULGARIAN
Czech	CZECH
Chinese-Simplified	CHINESES CHINESES-CAT CHINESES-X11
Chinese-Traditional	CHINESET CHINESET-CAT CHINESET-X11
Danish	DANISH
Dutch	DUTCH
Finnish	FINNISH
French	CFRENCH CFRE-X11-HELP FRENCH FRENCH-CAT FRENCH-X11 FRE-X11-HELP
German	GERMAN GERMAN-CAT GERMAN-X11 GER-X11-HELP
Greek	GREEK
Hebrew	HEBREW
Hungarian	HUNGARIAN
Icelandic	ICELANDIC
Italian	ITALIAN ITALIAN-CAT ITALIAN-X11

Language	Fileset Name
-----	-----
Japanese	JAPANESE JAPANESE-CAT JAPANESE-MAN JAPANESE-X11 JEUC-CAT JEUC-MAN JEUC-X11 JEUC-X11-HELP JPN-X11-HELP
Korean	KOREAN KOREAN-CAT KOREAN-X11 KOR-X11-HELP
Norwegian	NORWEGIAN
Polish	POLISH
Portuguese	PORTUGUESE
Rumanian	RUMANIAN
Russian	RUSSIAN
Serbocroatian	SERBOCROATIAN
Slovakian	SLOVAKIAN
Slovene	SLOVENE
Spanish	SPANISH SPANISH-CAT SPANISH-X11 SPA-X11-HELP
Swedish	SWEDISH
Thai	THAI
Turkish	TURKISH

Note You can also remove the MKFONTDIR fileset if you use neither graphics nor Asian languages: Japanese, Simplified Chinese, Traditional Chinese or Korean.

You must remove the filesets listed on the previous pages in order to benefit from choosing a “non-graphics” bundle as described in the previous sub-section, “Series 800 Small System”.

Proceed with caution when removing filesets.

- *Do not* remove an NLS fileset, and do not choose a “non-graphics” bundle, if you use or develop applications in the language in question.

(But you should still be able to remove the filesets for the languages you don’t use).

- *Do not* remove the graphics filesets, and do not choose a “non-graphics” bundle, if you use graphics or multimedia applications such as screen capture or SharedPrint, if your Series 800 runs Xterminals or acts as an HPVue or X server, or if you use NetLS.
- Always use `rmfn` to remove filesets; this ensures that you do not remove filesets that are required by other filesets that you need.

What To Do Next

The sections that follow outline other tasks you may need to do before upgrading to 10.01. Read the sections, do whatever tasks apply, then run `snoop` again to check that you have solved all **PROBLEMS** reported in the logfile.

Preparing for I/O System Changes

This section describes tasks you may need to do to prepare your system for the I/O-subsystem changes that were introduced at 10.0.

Before You Start

You should already have read the section on “I/O Convergence” in the *Release Notes for HP-UX 10.10*. If you haven’t, read it now. (See “Locating and Loading Tools and Documentation”, earlier in this chapter, for instructions for getting this document from tape or CDROM.)

You may also need the 9.0 versions of the Series 700 and 800 *HP-UX System Administration Tasks* manuals.

You should also make sure you have copies of the 10.0/10.01 versions of the following books:

- *Configuring HP-UX for Peripherals*
- *HP-UX System Administration Tasks*

If you have developed in-house drivers for any of your Series 700 systems, you also need the *Driver Development Guide*.

Summary of Tasks

As you can see from the *Release Notes for HP-UX 10.10*, most of the work for I/O convergence will be done for you during the upgrade itself. This section summarizes what preparation (or, in some cases, follow-up, after loading 10.01) you need to do, if any. Instructions are in the sections that follow; use this section as a checklist.

■ For *disks*:

- If you are running *DataPair/800*, order and migrate to *MirrorDisk/UX* (Series 800).
- Convert Series 800 disks from hard partitions (Series 800) to Logical Volume Manager (LVM), if you so decide.

■ For *drivers*:

- Make sure that drivers for key applications will be available before you upgrade to 10.01.
- If you use an optical autochanger, modify scripts and procedures to take account of the 10.01 driver change that allows you to mount only as many filesystems in the autochanger as the autochanger has drives.

■ For *device files*:

- Check code and scripts for use of `mknod` to create device files.
- Check for symbolic links.

The sections that follow provide instructions for these tasks, or tell you where to find instructions.

Disks

You should already have read the section on “Disk Configurations in 9.x and 10.x” under “I/O Convergence” in the *Release Notes for HP-UX 10.10*. That section outlines the disk-management choices available to you as you upgrade from 9.x to 10.01; if you didn’t read it or don’t recall it, spend a few minutes reviewing it before you go on.

You have work to do before the upgrade to 10.01 if:

- You are using unsupported disks; or
- You are using DataPair/800 for disk mirroring (Series 800); or
- You want to convert hard-partitioned disks to Logical Volume Manager *before* you upgrade to 10.01 (Series 800).

Note

- If you want to convert disks on a Series 800 system to Logical Volume Manager (LVM), you can do so before or after upgrading the system to 10.01 or 10.10.
 - If you want to convert disks on a Series 700 system to Logical Volume Manager (LVM), you must first upgrade the system and boot it on 10.01.
 - **9.07 systems** must be upgraded all the way to 10.10 before you can convert disks.
 - On other Series 700 systems, you can convert disks on 10.01, 10.10, or later.
 - You cannot convert disks to LVM *during* the upgrade.
 - Instructions for converting disks to LVM on a 10.01 or 10.10 system (Series 700 or 800) are in chapter 3 of the 10.01 version of the *HP-UX System Administration Tasks* manual, under “Converting Current Disks to New LVM Disks”.
-

If You Are Using Unsupported Disks

If the disks you are using came packaged with the system, or you bought them from HP to use with this system, they are supported; you need not read any more of this subsection.

If you have substituted or added a disk that you did not buy from HP:

■ If this is the *root* disk (the disk containing the */* directory):

- The upgrade could fail, even if the disk has been working on 9.x.

Call HP to confirm that this disk is supported for 10.01.

■ If this is not the root disk:

- Run `ioscan`.

- If `ioscan` shows that the disk is `CLAIMED`, the disk will upgrade successfully and will be accessible on 10.01.

- If `ioscan` does not show the disk as `CLAIMED`, the disk *will not* upgrade successfully and will not be accessible on 10.01.

(But you could still upgrade the rest of the system if decided to go ahead; an unrecognized non-root disk will not in itself cause the system upgrade to fail).

If You Are Running DataPair/800 (Series 800)

DataPair/800 is not supported as of 10.0. If you want to continue to use mirrored disks in 10.x, you need to:

1. Order MirrorDisk/UX from HP (optional product supported on 9.0 and later).
2. Migrate the disks to mirrored logical volumes, using LVM and MirrorDisk/UX.

The Series 800 *HP-UX System Administration Tasks* manual for 9.0 (HP part number B3108-90005) contains instructions for migrating disks mirrored via DataPair/800 to LVM and MirrorDisk/UX. See the section “Migrating Data From DataPair/800 to LVM’s MirrorDisk/UX” in chapter 8 of that manual, “Managing Logical Volumes”; the procedure begins on page 8-139.

4-60 Pre-Upgrade Tasks for All 9.x Systems

If You Want to Convert Hard-Partitioned Disks to LVM (Series 800)

10.x does not require LVM; existing hard-partitioned disks will continue to be supported.

But you may want to convert your disks to LVM—for example, if you are currently using DataPair/800 and want to continue using disk mirroring under 10.01, you will need to convert the disks in question to LVM and migrate to MirrorDisk/UX (see the previous section).

If you decide to convert some or all of your disks to LVM now (while running 9.x), follow the directions below.

There are three kinds of conversion you may want to do:

1. Convert hard-partitioned, non-root disks to LVM.
2. Move the root filesystem from a hard partition to a new disk to be managed via LVM.
3. Convert the current root disk to LVM.

1. To convert hard-partitioned non-root disks. Instructions for this are in the 9.0 version of the Series 800 *System Administration Tasks* manual, in chapter 8, “Managing Logical Volumes.” The procedure begins on page 8-134.

2. To move the root filesystem from a hard partition to a new disk to be managed via LVM. Instructions for this are in the 9.0 version of the Series 800 *System Administration Tasks* manual, in chapter 8, “Managing Logical Volumes.” The procedure begins on page 8-121.

3. To convert the current root disk to LVM. You need to back up the root disk, run the `lvmmigrate` tool, re-install the current version of HP-UX (for example 9.0), and recover the backup.

The following is a cookbook for doing this.

Note If you want a more detailed procedure, you'll find one in the Series 800 version of *Installing and Updating HP-UX 9.0*, in chapters 3, 4, 5 and 8; but note that you *do not* need to perform the steps in those chapters that deal with saving off and restoring custom files (you will simply back them up and recover them along with everything else) or loading the `lvmmigrate` tool (it is part of HP-UX 9.0, which is already on your system).

Perform the following steps.

1. Bring the system to single-user mode.
2. Do a full backup of the root disk.

(For backup procedures using various tools, see chapter 9, “Backing Up and Restoring Your Data”, in the 9.0 version of the Series 800 *System Administration Tasks* manual.)

3. Run `lvmmigrate(1M)`:

```
/etc/lvmmigrate -v | more
```

`lvmmigrate` saves its output in `/tmp/LVMMIGRATE`. Print out this file.

For a guide to using this file to prepare for re-installation, see the Series 800 version of *Installing and Updating HP-UX 9.0*, near the end of chapter 3.

See also the `lvmmigrate(1M)` manpage.

4. Re-install the current version of HP-UX (for example 9.0).
 - a. Shut down your system and turn off the computer.
 - b. Load the install tape or disk.
 - c. Boot the computer, interrupting the boot sequence so as to interact with the Initial System Loader (ISL).
 - d. Proceed with the installation, following prompts to install with LVM.

For detailed instructions, see chapters 4, 5, and 8 of *Installing and Updating HP-UX 9.0*.

Note You *do not* need to take the special steps indicated in chapter 8 of *Installing and Updating HP-UX 9.0* to recreate system files; simply recover them from your backup along with everything else when the migration is complete.

5. Migrate any filesystems, other than `root (/)` and `/usr`, that were on your root disk.

Use the `/tmp/LVMMIGRATE` file to identify the filesystems if you need to. Use SAM to create the necessary logical volumes and add back the filesystems.

See the section “Migrate Non-Root File Systems on Root Disk” in chapter 8 of *Installing and Updating HP-UX 9.0* if you need detailed instructions.
6. Recover your backup.
7. Return the system to multi-user mode to let your users back on.

Other Considerations for Series 800s

At 10.0, the definitions of sections 2 and 0 on hard-partitioned disks were switched, so that section 0 specifies “whole-disk” access and section 2 specifies the section of the disk that used to be section 0.

There is nothing specific you need to do to prepare for this: *no data will be moved or overwritten.*

Symbolic links will ensure that you continue to access the disk in the same way as you did before the upgrade. If you were accessing the whole disk via section 2 at 9.x, you will continue to read and write to the whole disk on 10.x. Disk access to 9.x section 0 will be redirected to the same physical location via 10.01 section 2.

Note

- If you keep diagrams mapping sections of your disks to their contents, you may want to switch the zeroes and two's before upgrading to 10.01.
- Most device-file names and major and minor numbers, including those for disk sections, are changing in 10.01, but you will still be able to address the correct section by the old name, because the old name will be symbolically linked to the new file.
- Once you are on 10.01, *do not* use `mknod(1M)` to create device files for disk sections.

Avoid `mknod` in general; it requires you to know major and minor numbers and to supply a device file name; since these are all changing in 10.01, you should use `mksf(1M)` and `insf(1M)` instead; they will do the right thing.

Considerations for Series 700s

For Series 700 machines, there are no disk-management tasks to do before you upgrade to 10.01. But keep the following points in mind:

- If your system includes *SDS disk arrays*:
 - During the upgrade to 10.01, the upgrade software will convert these arrays to striped logical volumes.

The conversion is done for you; you do not need to intervene to make it happen.

The following restrictions apply to the LVM configuration that the upgrade program creates:

- The volume group created for each SDS array can support a maximum of eight disks.

You can add disks to the volume group if the SDS array contained fewer than eight disks, but the group can never contain more than eight.

Note The manpage says a volume group can contain up to 16 physical volumes; but the volume group created by the SDS conversion is an exception.

This restriction does not apply if you create a new volume group.

- The volume group created for each SDS array can contain a maximum of eight logical volumes.

Note The manpage says a volume group contain up to 255 logical volumes; but the volume group created by the SDS conversion is an exception.

- HP does not at present support mirroring of data on LVM disks that have been migrated from SDS.
- LVM disks migrated from SDS do not support bad block relocation.

- If your system includes single *SDS striped disks*:
 - 10.x will continue to support these SDS disks via a “compatibility driver” (`cpd`).

During the upgrade, the disk label is rewritten so that these disks can be managed by the `cpd`.

Note

- As a result, these are no longer true SDS disks; they are more like Series 800 hard-partitioned disks.
- If the disk crashes, or the label is removed or damaged, you cannot recreate the label, nor replace the disk with another SDS disk.

In that case, the only way to continue to manage the disk's data in “stripes” is to reformat the disk, or its replacement, as an LVM striped disk. See the section on “Troubleshooting an Existing SDS Disk” in chapter 3 of the 10.01 version of the *HP-UX System Administration Tasks* manual.

-
- If you prefer to manage these disks under LVM, you can convert them to striped logical volumes after upgrading to 10.01. See “Converting Disks to LVM (Series 700)” in Chapter 7 for more information.

If there are any SDS disks on the system, the `LVM-RUN` fileset will be installed automatically during the upgrade (so long as you choose `Match What Target Has` and `DO NOT` “unmark” the `LVM-RUN` fileset). You can remove the fileset after the upgrade if your 9.x system did not have any SDS arrays; see “Removing LVM” in Chapter 7.

Caution

- Device files for all SDS disks must be in the path `/dev/dsk/` and they must be named according to the HP-UX 9.x naming convention for Series 700 disks.

See the next subsection, “Device File Convention Supported for SDS conversion”, for details.

- The 9.x-to-10.01 upgrade is the only way to convert SDS disks automatically.

On a 10.x system:

- An SDS array that has not been converted during the upgrade will not be usable.

SDS disks *will not* be converted during the upgrade if:

- They are attached to a cluster client that was not booted to the cluster during the upgrade to 10.01.
- They are not configured into the system to which they are attached before you run the upgrade to 10.01.
- They are not physically attached to the system and online during the upgrade.
- You do not load LVM during the upgrade to 10.01.

Match What Target Has selects LVM for you; make sure you choose **Match What Target Has** and **DO NOT unmark LVM**.

If you discover that you have accidentally failed to make an SDS array available for conversion, your only remedy is to attach it to another 9.x system, import it into that system (using `sdsadmin` with the `-i` option), then upgrade that system to 10.01.

- An SDS single disk can be converted to LVM (whether or not it has been converted for the `cpd` during the upgrade); but you must back up the data and re-initialize the disk for LVM. See “Converting Disks to LVM (Series 700)” in Chapter 7.

- If your system includes disks being managed as “whole disks” (the default until now):
 - You can continue on 10.01 with no change from 9.x; or
 - You can convert some or all of your disks to LVM after upgrading to 10.01.

See “Converting Disks to LVM (Series 700)” in Chapter 7 for more information.

Note**Root disk special cases:**

- The only way to convert a Series 700 root disk to LVM is to upgrade to 10.x, then re-install 10.x with LVM; conversion during the upgrade is not possible.
 - **9.07 systems** must be upgraded all the way to 10.10 before you can re-install with LVM. *Never try to re-install (or install) 10.01 on a 9.07 system.*
 - On other Series systems, you can re-install with LVM any time after upgrading to 10.01.
- If your root disk is currently configured for SDS, you have an unsupported configuration.

Back up your system and re-install the 9.x release you are currently on before attempting to upgrade to 10.01.

Device File Convention Supported for SDS conversion

The SDS conversion software (which is part of the 9.x-to-10.01 upgrade software) recognizes only device files whose names have the same format as those created during system installation time.

Device files whose names do not follow this convention will not be recognized, even if they were created by HP utilities such as SAM.

The following are examples of device files (in `/dev/dsk`) that will be recognized by the conversion software.

Type of Card	Device-File Name	Example
Single-ended SCSI	c<XXX>d<disk_addr>s0 where <XXX> is the high 3 nibbles of the minor number, in hexadecimal	c201d0s0 (for built-in SE-SCSI with disk_addr 0)
EISA	c4[1234]d<disk_addr>s0	c43d0s0 (for EISA card in slot 3 with disk_addr 0)
Fast/Wide SCSI:		(for Fast/Wide SCSI in slot 1)
9.05	c[ABCDEF]0ds0[ABCDEF]s0	cA0dB0s0
9.07	c[abcdef]0d[abcdef]s0	ca0dbs0

Note For Fast/Wide SCSI, if *disk_addr* is greater than 9, the disk address will be represented in hexadecimal upper case for 9.05 and in hexadecimal lower case for 9.07.

If you have SDS device files that do not meet these criteria, **snoop** will warn you that you need to create new ones. You *must* do this before you upgrade.

The following is an example of a warning message from **snoop** for a single-ended SCSI disk which has “201” as the minor number:

```
NOTE:    Found SDS disk /dev/dsk/c201d0s0
PROBLEM: upgrade has detected a SDS disk on your system.
         but the corresponding device file
         /dev/dsk/c201d0s0 does not exist. Please create
         the device files (both block and raw) for this disk.
         example:
         mknod /dev/dsk/c201d0s0 b <dev major number> 0x201000
NOTE:    Found SDS disk /dev/dsk/c201d1s0
```

Drivers

When you upgrade to 10.01, the following things will happen:

1. The upgrade software will remove all 9.x drivers from the kernel.
2. Drivers bundled with the 10.01 software being loaded onto your system will be bound into the new 10.01 kernel.

This means that the following classes of driver will *not* be bound into the kernel from which the new 10.01 system boots:

- Custom drivers written by your in-house programmers.
- Drivers written by third-party (non-HP) vendors.
- HP drivers that are not bundled with 10.01 (or that are bundled with 10.01 products that you haven't loaded).

What You Need to Do

- For in-house drivers (if any):
 - Warn programmers responsible for any in-house drivers that they should begin revising their code to comply with 10.x.

The *Release Notes for HP-UX 10.10* provide a summary of the changes needed; for details, programmers writing for the Series 700 will need the *10.0 Driver Development Guide*. HP is not publishing any guidelines for writing Series 800 drivers for 10.x.
- For third-party (non-HP) drivers:
 - Check with the vendors to find out when 10.x-compliant drivers will be available.
- For HP-supplied drivers that do not come bundled with HP-UX:
 - Check with your HP Service Engineer (SE) to find out when 10.x-compliant versions of these drivers (or the applications to which they belong) will be available.
 - For the autochanger:
 - Modify scripts and procedures, if necessary, to take account of a restriction in the 10.x driver, `ssfrc`, that allows you to mount only as many filesystems in the autochanger as the autochanger has drives.

See “Autochanger Driver Changers”, later in this section.

Are all 9.x drivers incompatible with 10.x? Not necessarily; in fact most drivers written according to guidelines in the 9.0 version of the *Driver Development Guide* should continue to work on 10.x. But HP cannot guarantee that they will work, which is why they will not be bound into the kernel built by the upgrade software, the kernel from which your 10.01 system will initially boot.

After booting 10.01 or 10.10, you can try rebuilding the kernel so as to include 9.x drivers that have been removed and not replaced during the upgrade (see “Adding Drivers” in Chapter 7).

But, for drivers that critical applications depend on, it is not a good idea to wait and see. Contact the supplier, find out when a 10.x-compliant version of the driver (or the application it is part of) will be ready, and wait until that time to upgrade.

Autochanger Driver Changers. Autochanger driver behavior has changed as of 10.0. On 10.x systems:

- You can mount only as many platters as there are actual autochanger drives.
- HP-UX creates only as many device files as are needed to communicate with those drives.
- Hard-partitioning on autochanger surfaces is no longer supported,
- The device-file naming convention has changed.

9.x autochanger device files will be removed during the upgrade to 10.01, and only those that relate to actual drives will be replaced (see “Device Files” later in this chapter).

For more information, see the manual *Installing and Administering Optical Jukeboxes on HP 9000 Series 700 and 800 Systems*; the driver changes are described in an appendix, “HP-UX 10.0 Autochanger Driver Changes and Developer’s Guide”.

Note

Order *Installing and Administering Optical Jukeboxes on HP 9000 Series 700 and 800 Systems* using the HP part number 5960-7624 Edition 3.

The manual title and part number given in earlier versions of *Release Notes* (before 10.10) are incorrect.

You’ll find a brief summary of the most important change, the restriction on virtual mounts, in the “I/O Convergence” section of the *Release Notes for HP-UX 10.10*, under the heading “Autochanger Virtual Mounting”.

The restriction on mounts may affect scripts. For example, in 10.x, `mediainit`, `newfs` and `mount` open a device with the `O_NODELAY` (non-blocking) flag set. If the `ssfrc` driver receives such a request when all the drives are busy, it will fail with `EBUSY`.

The following script, for an autochanger containing two drives, presents no problems on 9.x, but needs to be modified for 10.x:

```
#!/bin/ksh
# mi.ksh: 9.0x script to mediainit the first six
# cartridges in the autochanger
#
mediainit /dev/rac/1a &
mediainit /dev/rac/1b &
mediainit /dev/rac/2a &
mediainit /dev/rac/2b &
mediainit /dev/rac/3a &
mediainit /dev/rac/3b &
mediainit /dev/rac/4a &
mediainit /dev/rac/4b &
mediainit /dev/rac/5a &
mediainit /dev/rac/5b &
mediainit /dev/rac/6a &
mediainit /dev/rac/6b &
wait
```

This script should be modified as follows for 10.x:

```
#!/usr/bin/ksh
# mi.ksh: 10.x script to mediainit the first six
#cartridges in the autochanger.
#The 'waits' are necessary because mediainit in 10.x
#performs opens with 0_NDELAY.
#
mediainit /dev/rac/c1t3d0_1a &
mediainit /dev/rac/c1t3d0_1b &
wait
mediainit /dev/rac/c1t3d0_2a &
mediainit /dev/rac/c1t3d0_2b &
wait
mediainit /dev/rac/c1t3d0_3a &
mediainit /dev/rac/c1t3d0_3b &
wait
mediainit /dev/rac/c1t3d0_4a &
mediainit /dev/rac/c1t3d0_4b &
wait
mediainit /dev/rac/c1t3d0_5a &
mediainit /dev/rac/c1t3d0_5b &
wait
mediainit /dev/rac/c1t3d0_6a &
mediainit /dev/rac/c1t3d0_6b &
wait
```

If the script were left unmodified from the 9.x version, the last ten `mediainits` would fail with `EBUSY`, because no drives would be available.

For more information, see the appendix “HP-UX 10.0 Autochanger Driver Changes and Developer’s Guide” in the manual *Installing and Administering Optical Jukeboxes on HP 9000 Series 700 and 800 Systems*, HP Part Number 5960-7624 Edition 3.

4-74 Pre-Upgrade Tasks for All 9.x Systems

Device Files

Most of the device files on your 9.x system will be removed and replaced during the upgrade to 10.0. This is to accommodate a new naming convention and a new major and minor numbering scheme that support **I/O convergence**, that is, a single convention for both the Series 700 and 800.

You do not have to do a great deal to prepare for this change; although old device files will be removed, their names will be preserved as symbolic links to the new files. This means that code and scripts that invoke the old names will continue to work on 10.x.

But you do need to check for three things:

1. Use of `mknod(1M or 2)`, in code, scripts, or written procedures to create device files. Use `mksf` or `insf` instead of `mknod` on 10.x.

Use the `prepare(update_aid)` tool to check for calls to `mknod`; see Chapter 5 for details.

2. Symbolic links to 9.x device files.

Make a note of any such links that may exist on your current system. You will need to rebuild them after upgrading.

3. Autochanger device files that do not relate to actual drives.

The upgrade process will remove all the 9.x autochanger device files, and replace them with only as many new device files as the autochanger has drives.

The old names that related to actual drives will be retained on 10.x as links to the new files; but names of device files that have no corresponding drive will not.

Preparing for File System Layout Changes

- If you intend to rely on **transition links**, as described under “Upgrade Options” in Chapter 3, you don’t need to make any modifications to code and scripts to prepare for the transition to the 10.0 filesystem. (Remember that transition links will be supported only for the lifetime of 10.x.)

But you should read the *HP-UX 10.0 File System Layout White Paper* so as to understand how the filesystem layout will change.

In addition, even though you may have decided not to change pathnames in code and scripts, you may need to make some changes to accommodate changes to commands and system calls; see “Preparing for Changes to Commands and Libraries” later in this chapter.

- If you intend to convert pathnames in code and scripts before you upgrade (the “Full Preparation” approach), follow directions in Chapter 5.
- After the upgrade, you will need to modify system startup and shutdown scripts, and `/etc/inittab`, to merge custom code from the 9.x versions back into the 10.x files.

See Chapter 7.

What Will Happen to Non-System Files?

- File systems, directories and files you have created directly under the root (`/`) directory will not be moved.

For example, a directory `/myapps` will not be moved or touched during the upgrade.

- `/users` is left intact.

The upgrade software creates an empty directory `/home`.

After the upgrade, you may want to move and reorganize some directories, including `/users`, to comply with the 10.0 paradigm (as described in the *HP-UX 10.0 File System Layout White Paper*), but you do not have to do this.

- Directories and files you may have created under HP-UX “system” directories such as `/bin` will be moved to reside under the new parent directory.

4-76 Pre-Upgrade Tasks for All 9.x Systems

For example, if you have created a directory under `/bin` (we'll call it `/bin/myBinDir`), the upgrade software will move this directory, along with the rest of the `/bin` directory to `/usr/bin`.

But scripts that look for `/bin/myBinDir` will continue to find it, because `/bin` is linked by default to `/usr/bin` on 10.01.

Note `snoop` will warn you if there's already a `myBinDir` in `/usr/bin`,
If you have not resolved this potential conflict by the time you upgrade, `upgrade` will combine the former `/bin/myBinDir` with the existing `/usr/bin/myBinDir`, resolving name collisions at the file level by appending a suffix to one of the files.
(For example, if there is a file `/bin/MyBindir/tcbin` that would overwrite an existing file `/usr/bin/MyBindir/tcbin`, the existing file might be renamed to `/usr/bin/MyBindir/tcbin.save`.)

- HP products bundled with the 10.01 release, such as C++, will normally be moved in compliance with the 10.0 filesystem layout.

See the product-specific release notes for more information.

Preparing for Changes to Commands and Libraries

As you can see from the “Commands and Libraries” section of the *Release Notes for HP-UX 10.10*, HP-UX releases 10.0/10.01 introduce a large number of changes to commands and libraries. In some cases, these changes will cause problems for code and scripts that invoke the commands and routines in question.

- If you have chosen the “Quick Start” approach described in Chapter 3, HP recommends you do the following:
 - Scan the “Commands and Libraries” section of the *Release Notes for HP-UX 10.10*, and Chapter 8 of this manual, to get an idea of what has changed.
 - If it looks as if many of your scripts and applications will be affected, use the analysis and conversion tools described in Chapter 5 to identify the specific lines of code you need to change, and make those changes.

You can ignore warnings about invalid pathnames, since you are relying on **transition links** to take care of those.
- If you have chosen the “Full Preparation” approach described in Chapter 3, turn to Chapter 5 now and follow the directions for analyzing and converting your scripts, source files and documentation.

Then come back to this chapter to finish up the remaining pre-upgrade tasks.

Preparing for Other Changes

Run Levels

Run levels for many functions change in 10.01. In many cases, the run-level needed for a given function increases by one; for example, run-level 4 is needed to initiate VUE. See chapter 1 of the 10.01 version of the *HP-UX System Administration Tasks* manual, under “Controlling Usage and Processes with Run-Levels”, for more information.

Check your code, scripts and written procedures for assumptions about run-levels that may not be valid on 10.x.

Root User’s Home Directory

`upgrade` modifies the `.profile` found in the root user’s home directory (whether or not it is `/`) to make it 10.01-compliant. `.profile` files in other directories are *not* automatically modified.

Note This means that if root’s home directory is not `/`, `/.profile` will not be modified.

You should check and if necessary modify non-root users’ `.profile` and other shell startup scripts using the tools described in Chapter 5.

Cluster For advice on how to manage the server’s root directory in an NFS Diskless cluster, see “Managing Root’s Home Directory” in Chapter 2.

Certifying Applications

Well before upgrading to 10.01, you need to make sure that your critical applications are supported on 10.x, and, if you are going to 10.10, that they are available for that release as well.

In-House Applications and Scripts

Whether or not you use the tools described in Chapter 5 to create 10.01 versions of your scripts and in-house applications, you still need to run your normal regression tests and certification procedures on these applications to qualify them for 10.01 or 10.10, just as you did when you upgraded from 8.x to 9.x.

If possible, do not wait until after you have upgraded a production system to run these tests; do the certification on an upgraded test machine, so that the applications are ready to run as soon as the production system upgrade is complete.

HP Applications

At the time this manual went to press, the following HP applications had been certified on 10.01, but only for systems that could be upgraded with the tools provided with the initial release of 10.01; that is, these applications may *not* work correctly in NFS Diskless clusters, systems upgraded from 9.07, and other systems listed under “New Upgrade Capabilities” in Chapter 1.

Contact your HP Sales Representative (SR) for more information.

- 100VG-ANYLAN
- C Development Bundle
- SoftBench
- COBOL Development Bundle
- DCE Products
- DIALOG
- DTC Manager/UX
- EISA Fibre Channel
- EISA MUX software
- ENCAP
- FDDI 9000/700

4-80 Pre-Upgrade Tasks for All 9.x Systems

FTAM/9000
GLANCE PLUS
HP C++
HP ENWARE
HP FORTRAN
HP JetDirect
HP PASCAL
HP PCS
HP Perfview Agent
HP OnlineJFS
LAN Manager/X
LAN Manager V2.2
MC/ServiceGuard
MirrorDisk/UX
Netware V3.12
NIH Development Bundle
OmniBack II
OTS/9000 (Series 700)
PEX/PHIGS/PowerShade
HP Process Resource Manager
RTIEX-NIO
SNA++
SupportWave Diagnostics
Switchover/UX
Token Ring 9000 EISA
Toolbox
UIM/X 2.6
VT3K
X.25/9000 LINK
X.400 Messaging Transport

Building an SD Depot

Before you actually upgrade your system or systems, you may decide you want to repackage the 10.01 software that HP supplies on tapes or CDROM disks. For example, you may need to distribute the 10.01 upgrade from a local system's disk to systems at a remote site; or you might need to do the reverse: package the upgrade by downloading it from a network server onto tape. SD makes it easy for you to do this.

Caution **Cluster:** Do not mix the HP-UX 10.01 “core” release filesets with products from an Application Release (AR) in the same depot. This could cause the upgrade to fail.

Setting Up a Depot on a 9.x System

To serve (distribute) HP-UX 10.x software over a network from a 9.x system, using it to upgrade other 9.x systems to 10.01, follow the instructions below.

Note Depots on CDROM and disk can be served (distributed) directly over a network; software on tapes cannot.

1. Log in as the root user.
2. If you have not already done so, load the SD tools from the “Upgrade Tools” tape or CD as described in “Locating and Loading Tools and Documentation”, earlier in this chapter.

Make sure that at least the SD-AGENT, SD-CMDS, SD-DATA, SD-MAN and UPG-DOC filesets are on your server system.

3. Link the tools files to their actual names.

SD tools are shipped with `.gui` or `.tui` extensions so as to allow `snoop` to run them in “GUI” (Graphical User Interface) or “TUI” (Terminal User Interface) mode.

- a. Copy the script that follows and save it in a file called `sw9`.

This script is also included in the `README` file in the `UPG-DOC` fileset.

4-82 Pre-Upgrade Tasks for All 9.x Systems

```

#!/bin/posix/sh
if [[ $# != 1 ]];then
echo "Usage: sw9 gui/tui (gui = use SD with X11, tui = use terminal/console)"
exit 1
fi

for sd_file in "swacl" "swconfig" "swcopy" "swdepot" "swinstall" \
"swlist" "swreg" "swremove" "swverify"
do
rm -f /usr/sbin/${sd_file} > /dev/null 2>&1
ln /usr/sbin/${sd_file}.${1} /usr/sbin/${sd_file}
done

```

b. Make the script executable:

```
chmod +x sw9
```

c. Run the script:

- To enable SD commands to run on a graphical display with VUE or X11 run

```
sw9 gui
```

- To enable SD to run on a terminal (e.g. vt100, hp2392) or a non-graphics console run

```
sw9 tui
```

Save this **sw9** script for later use.

4. Register the software on CDROM as a depot.

(Make sure the CDROM drive is mounted; you can use SAM or the **mount(1M)** command to do this.)

For example, to register a CDROM drive mounted at **/cdrom**:

```
/usr/sbin/swreg -l depot /cdrom
```

If you prefer, you can copy all or part of the contents of the CDROM to hard disk and use that as a network software depot. (It is better to copy too much than too little.)

For example, to copy all the software on a CDROM into a depot at **/usr/main_depot** and automatically register it:

```
/usr/sbin/swcopy -s /cdrom "*" @ /usr/main_depot
```

Or, using `swcopy` in interactive mode (using screens like those you see in `snoop`):

```
/usr/sbin/swcopy -i -s /cdrom
```

If the 10.01 Software Is on Tape

You cannot distribute software across the network from a tape drive as you can from a CDROM drive.

But you can use the SD `swcopy` command to copy a tape or tapes into a software depot on a server and then distribute software from that depot across the network. This is especially useful if you have several systems to upgrade.

Caution If you have a *set* of tapes (tapes labelled something like **Tape 1 of 3**, **Tape 2 of 3**, **Tape 3 of 3**, ...) then you *must* use the interactive (`-i`) mode of `swcopy`, as in the example that follows.

To copy software on tape at `/dev/rmt/0m` to a depot at `/usr/main_depot`:

```
/usr/sbin/swcopy -i -s /dev/rmt/0m @ /usr/main_depot
```

The program will pause if you need to change tapes. Bring up the “Logfile” while in `swcopy` to see the tape-change messages.

More examples

The first command in the example below copies all software (“*”) from the network source `myserver` to the target `/mnt1/depot`. The second command does the same thing except that it copies only the software specified in the file `/tmp/langJ`.

```
swcopy -s myserver.cup.hp.com:/release/s700_10.01_gsK/goodssystem \
"*" @:/mnt1/depot
```

```
swcopy -f /tmp/langJ -s hpclpep:/languages/gsj @:/mnt1/depot
```

The following example builds a tape from the depot created in the previous example:

4-84 Pre-Upgrade Tasks for All 9.x Systems

```
swpackage -x target_type=tape -s /mnt1/depot -d /dev/rmt/0m "*"
```

Note Building a depot on tape or disk is a good use of the capabilities of SD, but you are taking on some extra responsibility: if you build the depot incorrectly, or incompletely, and the upgrade fails as a result, HP will not treat this as an SD defect.

Changes for Upgrades via CDROM

In the past, it was possible for US and European customers to do a complete upgrade from a single CD. As of 10.01, you must load a second CD. (Asian customers will need more; see “Bundles” in chapter 4, “Other Operating System and Subsystem Changes”, in the *Release Notes for HP-UX 10.10* for more information.)

Multi-User Licences

The additional CD is needed to update your multi-user licence, plus any optional software you may have ordered. On a multi-user system, your users may not be able to log back in until you have loaded the multi-user licence from the second CD and rebooted the system.

LVM Mirroring

If you are running a Series 800 system with mirroring, you must load the 10.01 version of LVM mirroring from the second CD before using any LVM commands.

The commands will fail if you try to use them after loading the software from the first CD but before loading the second.

Upgrading Networking Products

If your system includes networking products, read this section.

During the upgrade from 9.x to 10.01, the contents of `/etc/netlinkrc` will be moved to files under `/etc/rc.config.d/` files.

If you upgrade your operating system before your networking products, you may see a report of a networking startup failure when you reboot; this would be shown in the **Configure LAN interfaces** item in the **HP-UX Start-up in progress** list that is displayed on the console while the system is booting. The networking startup failure may be due to an attempt to configure a non-existent link (one of your networking products that needs to be upgraded to the 10.01 version.)

If this happens, continue with the upgrade. If, after you have upgraded the networking products, you still see a networking startup failure, use the `lanscan` command to verify the configuration in SAM, and check the `/etc/rc.log` file for any failure statements.

Follow these guidelines to ensure a smooth upgrade of a networked system:

- Do not upgrade to 10.01 until 10.01-compatible networking products are available. If you upgrade the operating system without compatible networking products, you will lose network functionality. To recover that functionality, upgrade the appropriate networking products.
- If possible, upgrade your existing networking products *at the same time* as you upgrade to 10.01.
- Install any new networking products *after* you upgrade to 10.01.
- If you can't upgrade all of your networking products at the same time, upgrade the software/hardware products (for example, Token Ring, FDDI, Fiber Channel) before the software-only products (software products that rely on a separate network hardware product to function—for example, OSI Transport Services, Netware).
- *Do not* move or reposition any network adapter cards before or during the upgrade process.
- Stop all applications that are running over the network before you start the upgrade process.
- Contact the suppliers of your non-HP products for information on upgrading them.

Hewlett-Packard itself does not support the upgrade of non-HP products to 10.01.
- You may need to change device-file names, interface names, and NMID numbers once you have upgraded the system.
- If problems arise during the upgrade process, check the following files:
`/etc/rc.log`, `/var/adm/sw/snoop.log`, and `/var/adm/sw/swagent.log`.

Getting Ready to Upgrade

You are nearly ready to begin the upgrade itself.

You have two more tasks to do (or three if this is a Trusted System cluster):

1. Re-run `snoop` to make sure that there are no more **PROBLEMS** or **WARNINGS**.

Follow the procedure in “Running `snoop`”, earlier in this chapter.

2. Back up your system.

See the next section, “Backing Up Your System”.

Cluster

If the system you are about to upgrade is a Trusted System cluster server, you must “unsecure” it before you can upgrade it. There will be a message in the `snoop` log to this effect.

To “unsecure” the system, enter the following command on the server:

```
/etc/tsconvert -r
```

See ““Unsecuring” a Trusted System” for more information.

Backing Up Your System

Before you begin the upgrade, do a full system backup. You should back up your data disks and your system (root) disk, but you’ll use a different utility for the root disk.

Data Disks

Back these up using whatever utility you normally use to do routine backups. See “Recovering, Moving or Mounting Files from a 9.x System” in Chapter 8 for guidelines on recovering this backup onto a 10.01 system.

Cluster

Use the `-H` to back up all elements of context-dependent files (CDFs). This will allow you to recover the 9.x “DUX” cluster in the unlikely event of a catastrophic failure during the upgrade.

Root Disk

Use the new utility, COPYUTIL.

COPYUTIL creates a complete image of the disk. It is the best way to create a failsafe backup that you can use in the unlikely event of a catastrophic failure during the upgrade.

COPYUTIL is part of the “Support Media” kit which you should have received along with the “Upgrade Tools” package of which this manual is a part. The “Support Media” kit also comes with a manual, the *Support Media User’s Guide* (HP part number 92453-90010).

Note If you have directories containing operating system files mounted from a disk other than the root disk (for example the `/usr` directory) use COPYUTIL to back up those disks too.

Do the following:

1. Find the “Support Media” kit and the manual.

Turn to Appendix A of the *Support Media User’s Guide*.

2. Put the tape or CDROM containing the offline diagnostics into the drive.
3. Boot the system from the tape or CDROM.

At the ISL prompt, enter:

```
ode copyutil
```

4. Follow directions in Appendix A of the *Support Media User’s Manual* to make a backup of your root disk.

Caution

- If this is a Series 700 system, do not rely on your recovery system (made with `mkrfs`); it may not be adequate to recover from a crash during the upgrade. Use COPYUTIL.
- COPYUTIL is supported for all 9.x systems that can be upgraded to 10.01, but the other tools in the “Support Media” package are supported only on 10.0 and later systems.

If your system crashes during the upgrade, or will not boot, see “Recovery from a Crash or Catastrophic Failure” in Chapter 7 and the *Support Media User’s Manual* for recovery instructions.

What To Do Next

You are now ready to upgrade your system to 10.01. Turn to Chapter 6.

Converting Code and Scripts

This chapter explains how to use the analysis tools to check 9.x source code, scripts, and text for compatibility with a 10.x system.

The tools are primarily intended to check for incompatible usage in code and scripts, but you can also run them on documentation files.

Before continuing with the tasks in this chapter, you need to understand something about the changes the tools help you provide for, specifically, changes between 9.x and 10.01 to:

- Pathnames (see the *HP-UX 10.0 File System Layout* White Paper)
- Commands, libraries and system calls (see Chapter 8)
- Device file names (see “I/O Convergence” in the *Release Notes for HP-UX 10.10*)

Although you can run the tools on a 10.x system, this chapter assumes you will analyze and convert files on the 9.x system, then upgrade to 10.01 system. (This is the “Full Preparation” option described in Chapter 3.) The tools themselves assume they are running on a 9.0 or later system.

The tools are also useful outside the context of a 9.x-to-10.01 upgrade; use them whenever you need to move code and scripts from a 9.x to a 10.x system. Follow the guidelines under “Moving Files” in Chapter 8 when recovering files from a 9.x to a 10.x system.

Caution **If you are upgrading to 10.10:**

Although the tools will be of great help in preparing your code, scripts and text for the transition from 9.x, they *will not* catch changes to commands and libraries that have occurred since 10.01. You will need to check the *Release Notes for HP-UX 10.10* for these.

What If You Don't Run the Tools?

If you have chosen the “Quick Start” approach described in Chapter 3, you should not need to spend too much time on the tasks described in this chapter before you upgrade. Some time before the next major release of HP-UX, you will need to convert your code and scripts, but initially you can rely on **transition links** to protect you from pathname changes.

Transition links (sometimes referred to as “fast transition links” or “compatibility links”) are installed by default on all 10.x systems and will be in effect on your system when you come up on 10.01, unless you specifically choose not to install the `UPG-TLINK` fileset. They link HP-UX 9.x “system” pathnames to their 10.x equivalents, and they are intended to reduce the amount of conversion work you will have to do on 9.x code and scripts when you first come up on 10.01.

But there could be problems transition links will not solve. For example, if your code uses commands or system calls that are no longer supported or whose options or behavior have changed, or if it builds or makes reference to 9.x device files (other than “default” names such as `/dev/lp` which are still recognized by HP-UX), you will have to fix it. The tools described in this chapter will help.

What the Tools Do

This tasks in this chapter make use of four tools: `prepare`, `analyzer`, `fscanner`, and `fnlookup`.

- `prepare` is a “front-end” to `analyzer`.

It identifies files that are likely to require analysis by the `analyzer` tool, and, optionally, invokes `analyzer` to process the resulting list(s) of files.

- `analyzer` scans an ASCII file—shell script, makefile, source file, or text file—and identifies pathnames, commands, system calls and library routines that may not work correctly on 10.x.

Optionally, it creates a copy of the file, substituting 10.x absolute pathnames for 9.x absolute pathnames (but not attempting to change commands or system calls).

Note

- The tools never modify the original file.

But `prepare` will, at your option, create files, and may also slow your system down while it’s executing.

- Pathnames constructed by variable substitution will *not* be caught.

When `analyzer` finds a potential problem, such as a command whose options have changed, it writes a comment into the new (copied) file, or into a log file, or both. The comment indicates the nature of the problem and provides a unique reference to a text paragraph that describes the problem fully. (You can choose to have `analyzer` write out the paragraph to the log file.)

- `fscanner` finds and evaluates ASCII strings in file or group of files.

If you use the `analyzer` keyword, `fscanner` will display all of the messages that `analyzer` has added to a converted file. `fscanner` has a graphical user interface similar to the `elm` mailer.

- `fnlookup` maps 9.x absolute pathnames to 10.x absolute pathnames.

For complete descriptions of these tools, their options, and their output, refer to the `prepare(update_aid)`, `analyzer(update_aid)`, `fscanner(update_aid)`, and `fnlookup(update_aid)` manpages.

What the Tools Don't Do

`prepare` and `analyzer` are aids to conversion, and they will make the process go more easily and quickly, but they are not a “black box” into which you can put your code and have it come out guaranteed to work on 10.x.

They will catch known incompatibilities between the HP-UX 9.x and 10.x “core” operating systems—the commands, utilities and libraries that comprise the basis of HP-UX. This “core” *does not* include all of the subsystems commonly bundled with HP-UX; for example, not all networking services are covered, and changes to the graphics and X libraries are not covered (see the *Release Notes for HP-UX 10.10* for information on these subsystems). To get an idea of what is covered and what is not, you may want to look at the “databases” (actually ordinary text files) that `prepare` and `analyzer` use; if you've loaded the tools onto a 9.x system, look at `/upgrade/lib/CMDS_DB` and `/upgrade/lib/LIBS_DB`.

Use the tools in this package in conjunction with the *Release Notes for HP-UX 10.10*, and with your normal regression-testing and certification procedures, to get code and scripts ready to run on 10.01.

5-4 Converting Code and Scripts

Analyzing and Converting Your Files

This section contains directions for converting 9.x scripts, source code, and other files with the `prepare` and `analyzer` tools. Unless otherwise stated, procedures and examples assume you are running on a 9.x system.

Caution Do not try to convert system files or move them from a 9.x to a 10.x system. You are unlikely to succeed in converting these files for yourself “by hand;” the upgrade process will do this conversion for you during the upgrade.

In this context, “system” files are HP-UX files in the following 9.x directories and 10.x directories: `/usr/bin`, `/usr/sbin`, `/usr/lib`, `/etc`, `/sbin`, `/bin`, `/lib`, `/dev`, `/system`. `prepare` will bypass these directories.

Analyzing Files

`prepare` and `analyzer` examine files for potential problems, and, optionally, create a parallel set of files in which pathnames have been converted to their 10.x equivalents. This section deals with the process up to that point; the following section, “Converting Files”, explains how to complete the conversion.

Strategies for Running `prepare` and `analyzer`

Together `prepare` and `analyzer` are capable of searching your 9.x system for all files that may need to be modified to run on 10.x, scoping the work, logging potential problems file by file and line by line, and creating new files in which the pathnames have been converted and other problems are flagged. How many of these capabilities you will want to use depends on how your system is set up, and perhaps on the major purposes of the system.

Read through each of the “cookbooks” below and decide which one best suits you; the “cases” are intended to help you decide but they are not binding: any of the “cookbooks” (and particularly the first two) will work for any “case”.

Note `prepare` and `analyzer` will identify problems related to HP-UX pathnames and HP-UX “core” commands and libraries. They will not catch problems specific to applications and utilities, whether these are provided by a third party or by HP itself.

Case 1. If the system has many scripts, makefiles and application source files scattered throughout the directory tree (or if you think it may have), and you want to convert most of them to run on the 10.01 system, then you should run `prepare` using the two-stage recipe recommended in the manpage and outlined later in this chapter under “Two-Stage Cookbook for Running Prepare”.

You can see a short version of this recipe if you simply invoke `prepare` without options:

```
/upgrade/bin/prepare
```

An example of a system that fits this description would be a multi-user development machine.

Note `/upgrade/bin/prepare -h`
 runs a demo of `prepare`.

Case 2. If you know that most of the files you need to analyze are concentrated in a few directories on the 9.x system, and you know exactly which directories they are, then you may want to use or adapt one of the examples in the “One-Stage Cookbook for Running Prepare” below.

If you are the owner/administrator of a single workstation, your system may fit this profile.

Case 3. If you are really only interested in a few files, and especially if they are large, you may be better off running `analyzer` directly on each file, without the `prepare` front end. See “Cookbook for Running Analyzer (without prepare)” later in this section.

5-6 Converting Code and Scripts

Two-Stage Cookbook for Running Prepare

This cookbook is for systems best described under “Case 1” above—that is, there are many scripts, makefiles and application source files scattered throughout the directory tree (or you think there may be.)

This procedure works particularly well for this case because the first stage weeds out files that *do not* need to be modified for 10.01, so that in the second stage you are running `prepare` on lists of files that `prepare` has identified as likely to need at least one change. This cuts down overall processing time considerably.

This procedure assumes a 9.x system. It uses the two-stage approach (“Identification” and “Conversion and Analysis”) described in the `prepare` manpage. It is the most “automated” way to get started with `prepare`, and has the advantage of weeding out files that do not need changes.

Before You Start.

- You may want to add `/upgrade/bin` to your path so you don’t have to type in the full pathname for `prepare` each time.

Procedure.

1. Run `prepare` once without options to create the `.preparerc` file that `prepare` requires.

Answer `y` when asked if you want to create the file.

2. Review the `.preparerc` file.

This file is copiously self-documented. Check the defaults for file types and directories to be included and excluded, to make sure you get the results you want.

3. Run `prepare` on a directory specified by *Directory_Name*:

```
prepare -bvr Directory_Name
```

where:

- b tells `prepare` to do only a quick check of files to see if they need further analysis and conversion.
- v tells `prepare` to report progress to `stdout`.
- r tells `prepare` to examine *Directory_Name* and the directories under it, recursively.

This creates four lists of files containing the names of all of the files `prepare` has examined. The file names are sorted into in these categories:

- 1.SHELL.g shell scripts and makefiles
- 1.SRC.g source files
- 1.TEXT.g text files
- 1.ELSE.g unrecognized files

Each of the files on these lists will have been subjected to a quick check for potential 10.x changes. Files that look as if they will need at least one change are listed in the following files:

- 1.SHELL shell scripts and makefiles
- 1.SRC source files
- 1.TEXT text files
- 1.ELSE unrecognized files

4. Edit each of these lists (1.SHELL, 1.SRC, 1.TEXT and 1.ELSE), removing files you don't want `prepare` and analyzer to look at, such as backup copies.
5. Run `prepare` on each of the lists of files.

5-8 Converting Code and Scripts

Note

- `prepare` will duplicate not only your files, but also the original directory tree, under the output directory you specify.
 - If you do not have enough disk space to create copies of the files, omit `-m NEWDIR` from the commands shown below.
-

Use these commands to analyze the files on each list, creating new files under `NEWDIR`:

```
prepare -V -l+2 -L $ -S $ -f 1.SHELL -m NEWDIR
prepare -V -l+2 -L $ -S $ -f 1.SRC -m NEWDIR
prepare -V -l+2 -L $ -S $ -f 1.TEXT -m NEWDIR
prepare -V -l+2 -L $ -S $ -f 1.ELSE -m NEWDIR
```

where:

- `-V` tells `prepare` to report progress (at the file level) to `stdout`.
- `-l+2 -L $` creates a logfile `1.SHELL.log`
- `-S $` creates a summary file `1.SHELL.sum`
- `-f` specifies the name of the file which contains the list of files to be examined.
- `-m` creates a new version of each original file under `NEWDIR`.
The new version has 10.x HP-UX pathnames in place of 9.x pathnames, and contains comments on other potential problems, such as changed options to commands or system calls.
`-m` creates `NEWDIR` if it does not exist; if it does exist, old files are not overwritten by new ones. (Use `-M` instead of `-m` if you want old files to be overwritten.)

6. Examine the new files (those under `NEWDIR` in this example) and do the remaining work of converting them. You can use `fscanner` for this task. See the `fscanner(update_aid)` manpage for more information.

See “Converting Files”, later in this chapter.

One-Stage Cookbook for Running Prepare

This cookbook should suit systems best described under “Case 2” above—that is, the files you want to convert for 10.x are concentrated in a few directories on the 9.x system, and you know exactly which directories these are.

This procedure assumes a 9.x system. It uses only the second, “Analysis and Conversion”, stage described in the `prepare` manpage.

Before You Start.

- You may want to add `/upgrade/bin` to your path so you don’t have to type in the full pathname for `prepare` each time.

Note that `prepare` will duplicate not only your files, but also the original directory tree, under the output directory you specify.

For example, if the original files are `/patrick/bin`, and you specify `/patrick/10.0files` as the output directory, the modified copies will be saved in `/patrick/10.0files/patrick/bin`

If you prefer, you can make `prepare` create a subdirectory `hp_REL10_hp` under each directory that contains files to be analyzed; `analyzer` then writes out the new files to those directories. To do this, use a period (`.`) instead of a directory name, for example:

```
/upgrade/bin/prepare -pas -m . -S mysum -f mylist
```

This is useful if the original files are only in one or two directories. For example, if all of the originals were in `/patrick/bin` and you specify “.” for the output directory, the copies would be saved `/patrick/bin/hp_REL10_hp`

Procedure.

1. Create a list of files.

Use `find(1)` to create a list of files that is to be examined and converted. For example, to create a list of all the files in the directory `/patrick/bin` and save it in a file called `mylist` in the current working directory:

```
find /patrick/bin -fsonly hfs -type f -print >mylist
```

5-10 Converting Code and Scripts

Note The `-fsonly hfs` option is important: by default, `prepare` will not cross NFS mount points, meaning that you must run it on the NFS server. You can change this behavior if you have to; see “ADVANCED OPTIONS” in the `.preparerc` file (created in step 4) for directions.

2. Edit the list, removing files you don't want `prepare` and `analyzer` to look at, such as backup copies.

It's a good idea to remove compressed files and other binaries from the list as well, as `analyzer` will report these as errors.

Note Remove blank lines, including trailing blank lines, from the list. `analyzer` reports the following error for each blank line it encounters:

```
analyzer ERROR: input file not specified
```

3. Scan the `prepare` manpage.

This cookbook does not exercise all of the options; there may be others that suit you better. (For example, you can specify options for a particular file on the list; these override the command-line options for that particular file.)

4. Run `prepare` once without options to create the `.preparerc` file that `prepare` requires:

```
/upgrade/bin/prepare
```

Answer `y` when asked if you want to create the default file.

5. Review the `.preparerc` file.

This file is copiously self-documented. Check the defaults for file types and directories to be included and excluded to make sure you get the results you want.

6. Run `prepare`; for example:

```
/upgrade/bin/prepare -pas -m /patrick/10.0files -S mysum -f mylist
```

where:

- pas** tells `prepare` and `analyzer` to examine:
- absolute pathnames (**-p**)
 - system calls and routines in ASCII source code (**-a**)
 - commands and options in shell scripts (**-s**)
- m** tells the tools to write out copies of the original files, converting pathnames to their 10.0 equivalents and adding comments on other problems such as changed options to a command.
- `/patrick/10.0files` specifies the directory to which `analyzer` will write out the copies of the files.
- S mysum** specifies the summary file that `analyzer` will write out.
- f mylist** is the input list of files to be examined (see step 1).

7. Examine the new files (those in `/patrick/10.0files` in this example) and do the remaining work of converting them.

See “Converting Files” later in this chapter.

Note

1. `prepare` will create the specified output directory (`/patrick/10.0files` in the above example) if it does not exist.

If the output directory does exist, `prepare` will not overwrite files that are already there, unless you specify **-M** instead of **-m**. If you specify **-M**, existing files will be overwritten by new ones whenever the names match.

2. `prepare` duplicates the original directory tree under the output directory you specify.
-

Examples

1. Examining scripts, not source code; write out detailed comments in log.

```
/upgrade/bin/prepare -ps -m /patrick/10.0files -l4 -L mylog -f mylist
```

- Examine files listed in the file `mylist` in the current working directory (`-f mylist`).
 - Check commands and pathnames but don't check for system calls or library calls (`-ps`).
 - Write output files to subdirectories under `/patrick/10.0files` (`-m /patrick/10.0files`).
 - Write out detailed log to file `mylog` in the current working directory, including the text paragraphs that comment on specific problems found in the files (`-l4 -L mylog`).
2. Examining mix of files that includes source code as well as scripts, makefiles, and possibly text files.

```
/upgrade/bin/prepare -V -pas -m . -S mysum -f mylist
```

- Run in verbose mode (`-V`), reporting actions on `stdout`.
 - Do all available checks: pathnames, library calls, commands (`-pas`).
 - Write output files to subdirectories named `hp_REL10_hp` under directories that contain the original files.
 - Write summary to file `mysum` in the current working directory.
3. Rechecking files on which you have already run `prepare` once.

```
/upgrade/bin/prepare -V -pas -M . -S mysum -f mylist
```

- Same as previous example, but overwrite `hp_REL10_hp` directories created by previous run of `prepare` (using the same list of files in `mylist`).

Useful prepare Options: Quick Reference

```
prepare -g
    "get": Generates lists classifying files according
           to types (output in *.g files).
prepare -q
    "quick": Runs a quick analysis on the files in the *.g
             list and outputs another list (n.Shell, n.SRC etc.) of
             files that may require at least one modification for
             10.01.
prepare -b
    "build": Same as prepare -g -q
prepare -m
    "modify": create modified files in a save directory
             ("-m ." is special and results in a directory
             hp_REL10_hp). Note that the full pathname is
             duplicated beneath the save directory.
prepare ... -l loglevel -L logfile
    "loglevel": 1...4 (precede number with + to get
                 additional statistics of number of changes). -l is
                 ignored unless the logfile is specified with -L.
                 Level 2 is usually about right; use 4 if you want the
                 text that explains the changes to be printed in the
                 logfile.
prepare ... -v|-V
    "verbose": -V recommended
```

Cookbook for Running Analyzer (without prepare)

This cookbook is for those who fit “Case 3” above—that is, if you want to examine only one or a few large files, and don’t need the `prepare` front end.

- To run `analyzer` directly on a single file (for example a large piece of code), use a command such as:

```
/upgrade/bin/analyzer -pas -l2 -L analyzer.pdlog -m myfile.10.0 myfile
```

- To run `analyzer` on a list of files, you might write a script such as:

```
while read filename
do
  /upgrade/bin/analyzer -ps -l2 -L analyzer.pdlog -m$filename.10.0 $filename
  mv $filename.10.0 $2
done < $1
```

where:

- \$1 is the filename of the input list of files.
- \$2 is a directory where the 10.01 versions of the files will be saved.

Converting Files

By this point you should have a set of partially converted files, in which the 9.x pathnames have been converted to their 10.x equivalents, and other potential problems have been flagged with comments.

Note

By no means will all potential problems actually turn out to be things you need to fix, but you need to check all of them.

In particular, look out for:

- Use of `mknod(1M)` or `mknod(2)` to create device files, and device file references in general. The names and the major or minor numbers will probably not be valid for 10.x.
- HP-UX ("DUX") cluster options.
- Kernel generation using `regen` or `uxgen` (for the Series 800). These commands are not supported in 10.0; see the section "Kernel Convergence" in the *Release Notes for HP-UX 10.10*.
- `-F` in file system commands; the meaning has changed in 10.x.

Conversely, `prepare` and `analyzer` will not catch problems that are specific to applications and utilities that are not part of the core operating system.

A quick way to find comments added by `analyzer` is to print out the summary file created with the `-S` option to `prepare` (see "One-Stage Cookbook for Running Prepare"). A number greater than zero in the `#WARN` column means there's at least one line in the file in question that needs your attention. You can find analyzer's comment by searching for `#analyzer` in the output file. (Another method is to run `fscanner` using the `analyzer` keyword. Refer to the `fscanner(update_aid)` manpage for details.)

The comment lines point to text that describes the issue more fully. Look up this text either in the log file (if you chose log option `-14`) or in the appropriate database (`/upgrade/lib/CMDS_DB` or `/upgrade/lib/LIBS_DB`).

Comments tagged with a code beginning with the letter C (e.g. C_120) refer to messages in the CMDS_DB; comments whose code begins with the letter L refer to messages in the LIBS_DB.

These “databases” are in fact ordinary ASCII text files; use the number (e.g. C_120) to search for the message via `more` or your favorite text editor. (Make sure you are in read-only mode!)

Check the comments in all of the files, make the changes that turn out to be necessary, then upgrade your system. Once you are up on 10.01, you should still do the testing and certification you would normally do on a new release before you run these programs and scripts or make them available to users.

Example

The following is an excerpt from a script that warns the user if he or she is logging in to a member of an HP-UX (“DUX”) cluster:

```
#!/bin/ksh

.
.
.
if [ -r /etc/clusterconf ]
then
  if [ "$(getcontext | grep remoteroot)" != "" ]
  then
    echo client of `cnodes -alC | grep ROOT | cut -d" " -f1`
  else
    echo cluster server
  fi
fi

.
.
.
```

After running `prepare`, you’d find that the summary file reports several warnings (`#WARN`) for this file. Examining the copy of the file that analyzer has created, you’d find this section commented as follows:

```

if [ -r OBSOLETE</etc/clusterconf>OBSOLETE ]
#analyzer, Line 58: Above pathname requires manual evaluation.
then
  if [ "$(getcontext | grep remoteroot)" != "" ]
#analyzer, Line 60: Command "getcontext", OBSOLETE, see CMDS_DB entry C_089
#analyzer, Line 60: Command "grep", WARNING, see CMDS_DB entry C_092
  then
    echo client of 'cnodes -a1C | grep ROOT | cut -d" " -f1'
#analyzer, Line 62: Command "cnodes", OBSOLETE, see CMDS_DB entry C_045
#analyzer, Line 62: Command "grep", WARNING, see CMDS_DB entry C_092
  else
    echo Cluster server
  fi
fi

```

Note This is a fairly severe example, generated to show as much analyzer output as possible; you should not find many passages of real code as densely commented as this.

The reference to `/etc/clusterconf` (line 58) is the first problem. The file won't exist on a 10.x system, hence the `OBSOLETE` flag (and since there is no replacement for `/etc/clusterconf`, analyzer can't supply a 10.0 pathname.)

The next line invokes `getcontext`, which on 9.x is an HP-proprietary command. Searching for `C_089` in `/upgrade/lib/CMDS_DB`, you'd find:

```

getcontext#obsolete#NA#C_089#
  The utility getcontext(1) is no longer supported. This
  option supports HP-UX clusters, whose underlying technology
  has changed as of 10.x. If your version of 10.x includes
  NFS Diskless, see "NFS Diskless" in the Release Notes for
  more information on HP's much enhanced implementation of the
  industry-standard diskless technology. #

```

This indicates the underlying problem: HP-UX (“DUX”) clusters are not supported as of 10.0 (see “Converting HP-UX (“DUX”) Clusters” in Chapter 2 of this manual), and so none of the supporting commands, such as `getcontext` and `cnodes`, will be there (though there is an entirely separate SVID3 system call `getcontext`).

5-18 Converting Code and Scripts

The remaining messages in this example refer to `grep`, but the changes in question (concerning regular expressions) turn out not to affect this particular piece of code.

To run the script on 10.01, you would have to remove these lines or comment them out, and (once you are running NFS Diskless) find some other way of warning users that they're getting HP-UX from a remote server.

Looking up Pathnames

You may need help figuring out where files reside in the 10.0 (“V.4”) filesystem layout.

Use the `fnlookup` tool, which maps 9.x absolute pathnames to 10.x absolute pathnames and vice versa. For example,

```
fnlookup -o /bin/ls
```

returns

```
/usr/bin/ls
```

See the `fnlookup(update_aid)` manpage for more information.

Note

- `fnlookup` requires the full pathname: `fnlookup -o ls` won't find anything.
- `fnlookup` will be on your 9.x system if you loaded the tools as shown in Chapter 4; it is also supplied on 10.x systems as `/opt/upgrade/bin/fnlookup`.

Upgrading Your System from 9.x to 10.01

This chapter contains instructions for upgrading a 9.x system to 10.01. “System” means a standalone system or an HP-UX (“DUX”) cluster.

Note You *must* upgrade 9.x systems (including 9.07 systems) to 10.01 before you can upgrade them to 10.10.

Once the 10.01 upgrade is complete, turn to Chapter 7 for directions for follow-on tasks, including upgrading to 10.10.

Before you get here, you should have read at least chapters 2, 3 and 4, and done all of the preparation tasks in chapter 4 that apply to the system you are about to upgrade.

You should *not* proceed with the tasks in this chapter if:

- You have not done all of the preparation tasks.
See the checklist a few pages on to make sure.
- You intend to *install* 10.01 or 10.10.
You need a different manual, *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.
- You are upgrading from 10.x.
Use *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

What Happens During the Upgrade

What upgrade Does for You

In the process of upgrading your system to HP-UX 10.01, the `upgrade` utility will reorganize HP-UX “system” directories to comply with the 10.0 filesystem layout (see the *Release Notes for HP-UX 10.10* and the *HP-UX 10.0 File System Layout* White Paper). “System” directories include the kernel and supporting commands, fundamental networking services and Xwindows.

`upgrade` translates the fileset information in the 9.x directories `/etc/filesets` and `/system` to the format used by HP Software Distributor (SD), the 10.x successor to `update(1M)`. It does this for all products which have been installed on your system via `/etc/update`. After the upgrade, SD will know about all HP products, as well as your third party applications if they were originally installed (at 9.x or earlier) with `update(1M)`.

In addition, `upgrade` will convert and save kernel and other configuration information for HP-UX (“DUX”) Cluster clients that are booted to the server during the upgrade; it will take you one additional configuration step after the upgrade completes on the server to add the clients back into the new NFS Diskless cluster. (See “Upgrading HP-UX (“DUX”) Cluster Clients”, later in this chapter.)

What upgrade Will Not Do for You

`upgrade` will not move files that are not part of the fundamental HP-UX operating system, unless they are in (or in directories under) HP-UX “system” directories that are being moved (for example, if you have created files and directories under `/bin`, they will be moved to `/usr/bin`; `/bin` is a link to `/usr/bin` on 10.x systems).

Products whose files won’t be moved include products from HP as well as from third party vendors. For example, `upgrade` will not move HP-C++, nor will it move the files from your RDBMS vendor (e.g. Oracle, Sybase).

These application upgrades must be performed with special upgrade utilities, provided by HP in the case of HP products, and the third-party vendor in the case of third-party products.

6-2 Upgrading Your System from 9.x to 10.01

`upgrade` will perform only a minimal conversion for HP-UX cluster clients that are not booted to the server during the upgrade; see “If Clients Are Not Booted” in Chapter 2.

When Do You Have To Do What?

Most of your existing 9.x scripts and applications should continue to run after the upgrade as they did before, because HP has provided **transition links**, which are special symbolic links from the 9.x pathname to the 10.x pathname. For example, `/bin`, the 9.x path, is linked to `/usr/bin`, the 10.x path. See Chapter 3 for more discussion of transition links.

But even if you plan to rely on transition links (Option A in Chapter 3), it is wise to run the `prepare` and `analyzer` tools on shell scripts (remember your shell startup scripts `.login` and `.profile`) and source files before you upgrade: in addition to detecting pathname changes, `prepare` and `analyzer` point out HP-UX commands and functions that have changed in HP-UX 10.x. For example, `cat(1)` will have some new options, and `dscopy(1)` will not be included in HP-UX 10.01.

Directions for running `prepare` and `analyzer` are in Chapter 5.

You will need to make some changes as soon as you upgrade; others can be deferred. Scripts that use `dscopy(1)` will fail, for example; you’ll need to change them before or right after you upgrade. Scripts that use command pathnames such as `/bin/who`, on the other hand, will continue to work because of the transition link from `/bin` to `/usr/bin`.

- For information on tasks you need to do, or may need to do, after the upgrade to 10.01, see Chapter 7.
- For information on pre-upgrade tasks, see chapters 2, 3 and 4.

Caution	Do not begin the upgrade if you have not read these pre-upgrade chapters and done the tasks they describe.
----------------	---

Checklist of Pre-Upgrade Tasks

Before you upgrade, check that you have either done each of the tasks in the following table, or determined that it is unnecessary for this particular system.

Pre-Upgrade Checklist

✓	Task	For More Information See:
	Understand the major 10.01 changes	“Understanding the Changes” in Chapter 4
	Decide on your upgrade strategy	Chapter 2; “Upgrade Paths” in Chapter 3
	Change existing procedures	“Changing Existing Procedures” in Chapter 4
	Load upgrade and analysis tools	“Locating and Loading Tools and Documentation” in Chapter 4
	Run snoop and fix all PROBLEMS	“Analyzing Your System” in Chapter 4
	Prepare for changes to the file system layout	“Preparing for File System Layout Changes” in Chapter 4
	Remove S300/400 clients and clean up “DUX” clusters	“Preparing for File System Layout Changes” in Chapter 4
	Create swap-reclamation files for “DUX” clusters	“Reclaiming Device Swap” in Chapter 4
	Prepare for changes to commands and libraries	“Preparing for Changes to Commands and Libraries” in Chapter 4; Chapter 8
	Remove filesets you don’t need	“Removing Unneeded Filesets” in Chapter 4
	Set up a network server if you need to	“Building an SD Depot” in Chapter 4
	Back up your system	“Backing Up Your System” in Chapter 4

6-4 Upgrading Your System from 9.x to 10.01

Obtaining and Entering Codewords

There are two types of software on the “Product Media” (the tapes or disks containing HP-UX 10.01 and 10.10): protected and unprotected. To load protected software, you must enter an authorized codeword and Customer ID in the Codeword Entry Dialog screen of SD. For unprotected software, no codeword or Customer ID is needed (for example, the HP-UX 10.01 “Runtime Bundle” does not require a codeword and Customer ID).

You will have access only to those items that are unprotected or for which you have the proper codeword. Your codeword and Customer ID will appear on the certificate entitled *Codeword for Installing Software* or you will receive a certificate entitled *Codeword for Installing Software - Submittal Certificate* along with a *Codeword Request Form*. To obtain the proper codeword and Customer ID, return the *Codeword Request Form* to HP License Administration.

If you choose to run the SD utility *non-interactively*, you will be prompted for codewords and Customer IDs before SD is launched.

Examine *ALL* the packages that contain your products and the Software Certificate(s). Keep the certificates handy. If your certificate does not include a codeword and Customer ID (and you want to load protected software) you must obtain one from Hewlett-Packard.

Note

If you are loading HP-UX from CDROM disks, multi-user licences (to allow more than two users on the system) are on the *second CD*, which also contains optional products such as LVM disk mirroring.

This means that when your system reboots after loading the software on the first disk, users may report that they can't log back in; if this is a multi-user system, load your multi-user licence from the second CD before bringing the system back online.

This also applies to tapes if you are not a subscriber to upgrade services and are upgrading from multiple tapes.

Running the Upgrade

The following procedure is for an **interactive** upgrade, in which you respond to prompts and choose menu items.

Note You should normally use this procedure, but under certain circumstances (for example, if you have to upgrade several similar systems, and this is not the first system to be upgraded) you may prefer to run the upgrade unattended: see “Running an Unattended Upgrade” later in this chapter.

Pre-Flight Checks

Have you:

- Backed up your system?
- Booted all cluster clients (or all you intend to upgrade) to the server?
- Created a swap-reclamation file for a cluster server if necessary?
- Connected, powered up and configured all SDS disks?
- Stopped all cluster applications?
- “Unsecured” the cluster if it was a 9.x Trusted System?

Upgrade Procedure

To upgrade your system to 10.01, do the following:

1. Log in as root.

If possible, do this on a system that has a graphical display, allowing you to use the “windows” features of SD.

Cluster

If you are upgrading an HP-UX (“DUX”) cluster, log in to the *cluster server*.

You can be logged in on the console or remotely from a system that is not part of the cluster, but you *must not* be logged in from a cluster client.

If you are going to run the upgrade from VUE on the cluster server’s console, log in *to VUE* as root user; otherwise your non-root windows will disappear during the upgrade.

2. If this is a standalone system (not a cluster server) bring the system to single-user state:

```
/etc/shutdown
```

Cluster

If this is a cluster server:

- Make sure all users are off the system and bring down all applications, but *do not* shut the system down to single-user mode.

The clients need to stay up for the first phase of the upgrade.

- Check that all the clients are booted to the cluster:

```
cnodes -a
```

Clients listed with an asterisk are *not* booted to the cluster. `upgrade` will refer to these as **offline** clients in messages and they will not be fully converted to NFS Diskless (see “If Clients Are Not Booted” in Chapter 2).

Make sure all the clients you intend to upgrade are booted to the server now. If you bring more clients online once the upgrade has started, `upgrade` will not recognize them; you will need to restart `upgrade` to get it to recognize these clients.

`upgrade` will copy and modify the `/.rhosts` file on the server (or create it if it doesn't exist) so that it can obtain information about each of the active clients via `remsh(1)`. Before the `upgrade` program exits, it will put your original `/.rhosts` back in place.

Caution

- If this is a Series 800 system running LVM:

- If you have `/usr` on a logical volume other than the root volume, you will need to remount `/usr` after running `shutdown`:

```
mount lvol /usr
```

- *Do not* remount NFS mounts.

`upgrade` will explicitly unmount them. See “Shared HP-UX “System” Directories” in Chapter 2 for a discussion of how to proceed if you have HP-UX “system” directories NFS-mounted (for example directories under `/usr`).

3. On Xterminals and Series 700 workstations, restart VUE if necessary, set the DISPLAY variable and open windows:

- Restart VUE:

If you are running the upgrade from the console of the workstation you are upgrading, and this is a standalone system (not a cluster server) you should have shut the system down by now, killing VUE.

You can restart VUE by entering,

```
/etc/vuerc
```

This allows you to run the SD portion of the upgrade in graphical mode (with windows).

- Set DISPLAY:

If you are running the upgrade from a remote workstation or Xterminal, set your DISPLAY variable, for example:

```
DISPLAY=hpulpcu2:0.0; export DISPLAY
```

This will allow the SD (Software Distributor) portion of the upgrade to run in graphical mode (with windows).

- Open a second window:

If you are running the upgrade from any workstation or Xterminal, open a second window:

```
/usr/bin/X11/hpterm &
```

This will allow you use a subset of HP-UX commands during the upgrade.

4. Unless the 10.01 software is to be loaded from a network server, put the first 10.01 tape (there may be only one) or CD in the drive.

If you are using a CD, make sure the CDROM device is mounted,

```
/etc/mount
```

If there is no entry for the CD-ROM device, mount it; for example:

```
/etc/mount /dev/dsk/2s0 /cdrom
```

6-10 Upgrading Your System from 9.x to 10.01

5. Run the upgrade program:

```
/usr/sbin/upgrade
```

This starts `upgrade` in its default interactive mode.

6. `upgrade` asks you if you can use the Graphical User Interface.

Respond `y` if you're using a graphical display and set the `DISPLAY` variable as recommended in Step 3 of this procedure.

Troubleshooting

Problem	What to do
FATAL ERROR: X Toolkit Error: Cannot perform malloc	Your system is short of memory. Try running the upgrade from a remote system. If you have less than 16Mb of memory, you <i>cannot</i> upgrade to 10.01. If you continue to get this error after moving to a remote system, it may mean that the system being upgraded is short of swap space.
(errors indicating X connection was refused)	If you are running the upgrade from a window on a remote machine, make sure that you have permission to connect to the system being upgraded: try entering <code>/usr/bin/X11/xhost +</code> or <code>/usr/bin/X11/xhost + hostname</code> , where <i>hostname</i> is the official hostname of the machine being upgraded.
ERROR: upgrade is already running on this system. Only one instance of upgrade may be run at a time.	You may have aborted <code>upgrade</code> in the past (or it may have terminated abnormally for some reason). If you are sure no other upgrade process is already running, remove <code>/etc/upgrade/save/info/upgrade.lock</code> and start again.

You'll see messages on the screen that filesets are being “packaged”; what's happening is that **upgrade** is creating packages similar to those in `/etc/filesets`, but recognizable by SD, so that SD can load the equivalent 10.01 software.

Note **upgrade** needs about 5Mb of space on your root disk for this and other operations during the upgrade.

7. **upgrade** now checks for conditions that will pose problems for the upgrade.
 - If you have followed directions in Chapter 4, there should be no such conditions.
 - If **upgrade** reports **ERRORS**, this system is not yet ready to be upgraded.

Caution **DO NOT GO ON.** Find the problem in `/var/adm/sw/swagent.log`, and then go back to “Handling snoop **PROBLEMS**” in Chapter 4 for suggestions on dealing with conditions that prevent upgrade.

8. **upgrade** now invokes the **swinstall** program (part of Software Distributor—we'll use the umbrella term **SD** from now on).

SD first looks for the 10.01 software under `/var/spool/sw` on your system's root disk; this is the default location for an SD software "depot".

("Depot" is SD's name for the software source from which you are going to upgrade; see "Building an SD Depot" in Chapter 4 for information on building a software depot.)

There will not be any such directory unless you have built a "depot"; so SD defaults to `/dev/rmt/0m` instead. You can accept this default *unless*:

- You have loaded the tape into a drive addressed by a device file other than `/dev/rmt/0m`.

Or

- You are upgrading from a network server.

Enter the official hostname and the software source "depot", for example:

```
Source:          upgserver
Source depot path: /release/s700_10.01
```

You can also select from available servers and "depots" by clicking on **Source** and **Source depot path** respectively.

Caution **Cluster:** For an HP-UX ("DUX") cluster, the depot must contain only the HP-UX core software, not HP Application Release ("AR") software.

Or

- You are upgrading from CD-ROM.

Enter the name of the filesystem to which the CD-ROM drive is mounted (the **mount point**) for example,

```
/cdrom
```

9. The **Software Selection** window comes up.

Pull down the **Actions** menu and choose **Match What Target Has**.

This means that you want to load a system that has functionality at least equivalent to your current, 9.x system (**target** refers to the system you're upgrading).

Caution

- Always choose **Match What Target Has**.

This is the only way to guarantee that you will load a version of HP-UX 10.01 that boots and runs correctly, and continues to perform the functions you relied on in 9.x.

- Do not “unmark” software that SD selects when you choose **Match What Target Has** (that is, software selections that have **Yes** or **Partial** under **Marked?** on the **Software Selection** screen after you choose **Match What Target Has**).

Software you may not recognize as something you need may be required by HP-UX; unmarking it could cause your system to fail to boot 10.01, or to run incorrectly.

For example, if you are upgrading an HP-UX (“DUX”) cluster server, the NFS Diskless software you will need on 10.x will be automatically selected for you when you choose **Match What Target Has**. If you unmark it, you will not be able to boot the cluster on 10.01.

You may also want some or all of the new features being offered in HP-UX 10.01. Do not select these new features now: just choose **Match What Target Has**, complete the upgrade as described in the steps that follow, then run SD again as described in “Loading New 10.01 Features” in Chapter 7.

See “What You Will See” in Chapter 4 for a full discussion.

10. When the **Software Selection** window re-appears, pull down the **Actions** menu again, and this time choose **Install (analysis)**.

This brings up the **Install Analysis** window, which gives you the option of reading the analysis logfile as it's being written: click on **Logfile** to see the logfile.

11. When the analysis is complete, you'll see a message such as **Ready** or **Ready with warnings** or **Ready with errors**.

Caution

- If there are errors, **DO NOT GO ON**.

The upgrade will fail. You must fix the problem; see “Handling snoop PROBLEMS” in Chapter 4 for guidance. Proceed as follows:

- a. Exit the program.
- b. Find the report of the error in `/var/adm/sw/swagent.log`.
- c. Take whatever corrective action is recommended in the logfile.
- d. Run the upgrade program again, starting at step 1 of this procedure.

- If there are warnings, check the logfile for details (click on **Logfile**).

You should not see any warnings you have not already seen and investigated when you were running **snoop**. It is safe to proceed to load the 10.01 software only if there are no errors *and*

- there are no warnings *or*
- you have already determined that these warnings will not prevent a successful upgrade.

Otherwise you must find and fix the problem indicated by the warning before continuing with the upgrade: exit the program, find and fix the problem, then re-run the upgrade, as described under the previous bullet.

12. Click on **Disk Space** to confirm that you have sufficient free disk space to load 10.01.

Caution

If you do not have enough disk space, **DO NOT GO ON**:

- a. Exit the program.
- b. Free up or add the amount of disk space shown in `/var/adm/sw/swagent.log`.
- c. Run the upgrade program again, starting at step 1 of this procedure.

See the disk-space discussion under “Handling snoop PROBLEMS” in Chapter 4 for more information and suggestions.

13. If you are sure there are no problems you need to fix, click **OK** on the **Install Analysis** screen.

You'll get a message that the 10.01 software is about to be loaded on your system. Click **OK**, and click **OK** again on the message that warns you about updates to the kernel.

14. The upgrade tools now convert the filesystem to the new layout and modify configuration scripts to conform to it, and SD loads 10.01 onto your system.

When you get a message that the upgrade is complete, click **OK** to reboot the system.

You'll get a message telling you to check the logfile `/var/adm/sw/swagent.log` to see if there were any problems, and you'll also be reminded that you need to merge custom code from 9.x startup and shutdown scripts into the equivalent 10.01 files. See Chapter 7 for more information on this.

Troubleshooting

Problem	What to do
Cannot fork to run <i>filename</i>	Your system is short of memory. Try closing all windows and running the upgrade from a remote system.
(SD reports products or filesets not found.)	If the 10.01 software is in an SD “depot”, check that you have built the depot correctly and loaded everything you need from the tapes or CDs supplied by HP. See “Building an SD Depot” in Chapter 4.
(SD reports many filesets skipped.)	Check that you are using the right software source: you may have accidentally loaded the wrong tape or disk, or connected to the wrong “depot” (for example, a Series 700 source for a Series 800 system). If you have already upgraded to 10.01, and are loading additional bundles, skipped filesets are probably ok; see “Loading New 10.01 Features” in Chapter 7.

15. The system will now reboot to 10.01.

This first boot will take a little longer than usual; this is because post-installation scripts are running to complete the conversion from 9.x to 10.01.

Cluster

If you are upgrading a cluster server, the `reclaim_swap` program will run automatically immediately after the system boots. Do not intervene.

`reclaim_swap` recovers device swap from the server's disk space and adds it back into the file system, using information you have provided in the file `/etc/upgrade/save/server/ttools/AllSwap.d/reclaim_swap` (see "Reclaiming Device Swap" in Chapter 4).

- If you have not created this file, no swap will be reclaimed; you'll see boot messages followed a message from `reclaim_swap` that the file was not found or empty.

This is *not* necessarily an error; it was your choice whether or not to reclaim swap.

- If you have created the swap-reclamation file, you'll see boot messages followed by messages from `reclaim_swap`, such as:

```
reclaim_swap: Starting phase 1 of 3.

Verifying that a required kernel patch is present...
Verification completed.
Verifying /etc/upgrade/save/server/ttools/AllSwap.d/reclaim_swap

/dev/rdisk/c0t2d0      45      OK
/dev/rdisk/c0t6d0     110     OK
Verification completed.

Note: Reducing size of LIF swap entry on /dev/rdisk/c0t2d0
Note: /dev/rdisk/c0t6d0 is not a LIF volume; no LIF changes needed

reclaim_swap: Phase 1 of 3 completed.

NOTE: THE SYSTEM WILL NOW SHUT DOWN AND REBOOT TO SINGLE-USER MODE.
      DO NOT INTERRUPT THIS PROCEDURE!
```

`reclaim_swap` will reboot the system twice. This procedure should run smoothly without intervention from you. Check `/etc/rc.log.old` for `reclaim_swap` messages.

If `reclaim_swap` should report an error, see "Troubleshooting `reclaim_swap`" later in this chapter.

When the boot completes, see `/etc/rc.log` for details of what the post-installation scripts have done.

You should copy `/etc/rc.log` to a another name at this point. It contains valuable information which will be overwritten the next time the system boots.

16. If you are upgrading from CD, or you are not a subscriber to upgrade services and are upgrading from multiple tapes, load the second tape or disk now and follow directions in the **Caution** that follows.

Caution

- If this is a multi-user system, load the multi-user licences *from the second tape or CD* now.

Otherwise your users may not be able to log back in.

- If this is a Series 800 system running LVM with mirroring, you must load LVM mirroring *from the second tape or CD* before using any LVM commands.

These commands will fail if you try to run them after loading the software from the first tape or CD but before loading the second.

Troubleshooting `reclaim_swap`

`reclaim_swap` runs only on cluster servers; if you are not upgrading a cluster server, skip this section.

Error Messages and What They Mean.

```
reclaim_swap: Insufficient pseudo-swap.  xxKB present;  
4096KB required.
```

Cause: There is not enough memory-based swap (pseudo-swap) to run the reclaim swap tools. The tools need some amount of swap space to execute. This could happen only on a minimum memory configuration with a very large number of drivers and/or large kernel tables (e.g. kernel parameters such as `nproc`, `nfile`, `ninode`, etc., set to large values).

Failed to locate LIF volume header on /dev/rdisk/xxx

or

No SWAP entry in LIF directory on /dev/rdisk/xxx

or

No File System entry in LIF directory on /dev/rdisk/xxx

or

SWAP cannot be reduced to below 1 swap chunk

or

Change of xx sectors would extend into File System area

followed by:

reclaim_swap: Aborting due to error

Cause: Internal problem with LIF information; or you may have tried to reclaim too much swap space on a device.

Note: Increasing size of LIF file system entry on /dev/rdisk/xxx

followed by:

Warning: could not update LIF entry

Cause: Internal problem with LIF information. This condition should not cause a problem with the space that has been reclaimed, but should be looked into: call HP.

6-20 Upgrading Your System from 9.x to 10.01

(long error message from extendfs followed by:)

WARNING: The file system on "/dev/rdisk/xxx" may be corrupted.

followed by:

reclaim_swap: Aborting due to error

Cause: the tool used to extend the file system failed for some reason. If the disk is the root disk, then the **phase 3** boot (as reported by **reclaim_swap**) will stop in **fsck** and ask you to repair the file system. Otherwise, run **fsck** manually on the disk. Either way, call HP, and if necessary follow directions under "Recovery from a Crash or Catastrophic Failure" in Chapter 7.

reclaim_swap: /usr/bin/uname is missing or is not executable

or

reclaim_swap: /usr/sbin/swapinfo is missing or not executable
Is the file system that contains /usr/[s]bin mounted?

Cause: probably an incomplete upgrade to 10.01; check the logfiles as directed in "Examining the Upgrade Log Files" in Chapter 7.

What To Do Next

Your system is now running 10.01. Log in.

Note To run VUE on 10.01, you need to be at run-level 4.

If this is not a cluster server or a Series 700 system that was running 9.07, read the **Caution** below and turn to Chapter 7 for information on the tasks you may need to do before your users log back in.

Caution Upgrade HP applications to 10.01 before updating the system to 10.10.

- Not all HP applications are supported on 10.01 and 10.10. For a list of those supported on 10.01, see “Certifying Applications” in Chapter 4.
-

9.07 If this is a standalone system running 9.07, you must now immediately add supported 10.01 applications and then upgrade to 10.10. The transitional 10.01 version your system is running at the moment is supported only for these purposes.

- If you have not already contacted your HP Sales Representative to make sure your critical HP applications are supported on systems upgraded from 9.07, do so *before* loading the 10.01 versions onto the system.

Turn to “Loading New 10.01 Features and HP Applications” in Chapter 7 for instructions on updating HP applications.

Cluster If you are upgrading a cluster server (**including 9.07 clusters**), you need to run another program, `upgdiskless`, before you can boot the clients.

See the next section, “Upgrading HP-UX (“DUX”) Cluster Clients”.

Upgrading HP-UX (“DUX”) Cluster Clients

At this point, the cluster server has been upgraded and configuration information has been saved for each client in the server’s directory `/etc/upgrade/save/nodename` (*nodename* is the name of the 9.x cluster client).

Note

- Information has been saved for all clients, whether they were booted to the server or not when you ran the upgrade, but most of the information for clients that were offline (down or not booted to this cluster) will not be used by `upgdiskless`.
See “Converting HP-UX (“DUX”) Clusters” in Chapter 2 for more information.
- 9.x configuration information for the server, including `/etc/clusterconf`, has been saved in directories under `/etc/upgrade/save/server`.

Before You Start

Before you get here, you should have read, at the very least, Appendix B of this manual and the section “Planning Your Cluster Policies” in chapter 11 of the *HP-UX System Administration Tasks* manual.

To be fully prepared, you need to read the *NFS Diskless Concepts and Administration* white paper. A postscript version is on your 10.01 system in the file `/usr/share/doc/NFSD_Concepts_Admin.ps`.

See “Understanding the Changes” in Chapter 4 for more information on required reading. The documents are now on your system in `/usr/share/doc`.

Procedure for Upgrading Clients

1. Reconfigure swap devices (disks dedicated to device swap) to contain file systems if `snoop` has warned you that this is likely to be necessary.

You need to make approximately 44Mb of file-system space available for each client under `/export`. See “Converting HP-UX (“DUX”) Clusters” in Chapter 2, and for information on juggling space on other volumes to make it available under `/export/private_roots`, see section 6 of the *NFS Diskless Concepts and Administration* white paper (there’s a slightly abridged version in Appendix B of the present manual).

2. Make sure you are logged in to the *cluster server* as superuser.
3. Run `/opt/upgrade/bin/upgdiskless`.
4. Respond to the prompts to set `SHARED` or `PRIVATE` policies.

`upgdiskless` is an interactive program similar to SAM (the HP-UX System Administration Manager program). It provides online help, and offers you defaults wherever possible.

The defaults are designed to make your NFS Diskless cluster behave as much like the 9.x “DUX” cluster as possible. Enter 9 to accept all the defaults, setting all policies to `SHARED`.

If this is not what you want, change some or all of the `SHARED` items to `PRIVATE`; for example, if you want to restrict each user’s login to his or her own workstation, enter 1 on the `NFS Diskless Policies` screen.

Caution You can’t change the policies once you have added clients to cluster, without first removing all the clients and starting over.

5. Select the clients you want `upgdiskless` to configure into the NFS Diskless cluster.

`upgdiskless` lists the clients five at a time and numbers them from 1 to 5 (under **Number** on the left of the screen). Select a client by entering its number.

Note This will always be a number between 1 and 5. Other numbers tell `upgdiskless` to perform actions. For example 6 means go to the next screenful (“page”) of clients, if any; and this next set of five clients will again be numbered from 1 to 5. From any “page” before the first, you can enter 0 to get back to the previous “page”.

6. Enter 9 to go to the **Install Clients** screen.

`upgdiskless` will check for disk and filesystem space problems.

7. If `upgdiskless` does not report any space problems, enter 1 on the **Install Clients** screen to configure the clients.

8. Boot the clients.

Turn power off and then on again on each client.

As the clients come up, `upgdiskless` will run the `upgrade` program on each of them to complete their conversion from 9.x “DUX” to 10.01 NFS Diskless clients, and then reboot the clients a second time.

By the second boot, the clients should be swapping locally if they were so configured when you began the upgrade.

See “Converting HP-UX (“DUX”) Clusters” in Chapter 2 for an account of what is converted. See `/var/sam/log/samlog` for a log of what `upgdiskless` has done.

9. Fine-tune the clients’ configuration.

Some functions of “DUX” cannot be duplicated (for example, device swap to another client’s disk). Others are available in NFS Diskless, but must be reconfigured manually. Use SAM to configure these functions.

For example, to make a non-system directory on the server visible to all clients, do one of the following:

- a. If the directory is a separately mounted volume (for example a disk mounted on `/users`), select it from the SAM **File Systems** menu on the server, and then select **Manage Cluster-Wide** from the pull-down **Actions** menu.
- b. If the directory in question is not a mounted volume (for example, if `/users` is a simply a directory on the root disk), export it on the server, then run SAM *on a client*, add the directory as a remote file system and select the **Manage Cluster-Wide** button on the **Add** dialog box.
- c. To make a file system on a client's local disk available to all cluster clients, run SAM on the client and select the file system from the **File Systems** menu, then select **Manage Cluster-Wide** from the pull-down **Actions** menu. (To distribute a directory that is not a separately-mounted volume, export it on the client, run SAM on a *different* client and choose **Manage Cluster-Wide**.)

Note

`upgdiskless` has saved 9.x configuration files for fully converted clients (those that were booted to the cluster during the upgrade and successfully converted to NFS Diskless) in directories under `/export/shared_roots/client/etc/upgrade/save` on the server (that is, `/etc/upgrade/save` on the client).

9.x configuration files for clients that were not fully converted (for example clients that were not booted to the server when the server was upgraded) remain under `/etc/upgrade/save/client` on the server.

What To Do Next

Your cluster is now up and running under NFS Diskless on HP-UX 10.01. Read the **Caution** on the next page and then turn to Chapter 7 for information on the tasks you may need to do before your users log back in.

Caution**■ Upgrade HP applications to 10.01 before upgrading to 10.10.**

□ Not all 10.01 and 10.10 applications are guaranteed to work correctly when distributed in an NFS Diskless cluster; if you have not already contacted your HP Sales Representative to make sure critical HP applications will work as you need them to, do so *before* loading the 10.01 versions onto the cluster.

■ 9.07: If the cluster server was running 9.07, you must now immediately load supported 10.01 applications and upgrade to 10.10. The transitional 10.01 version your system is running at the moment is supported only for these purposes.

□ If you have not already contacted your HP Sales Representative to make sure critical HP applications are supported on NFS Diskless systems upgraded from 9.07, do so *before* loading the 10.01 versions onto the cluster.

Turn to “Loading New 10.01 Features and HP Applications” in Chapter 7 for directions for updating HP applications to 10.01.

Running an Unattended Upgrade

If you are upgrading several similar systems, you may not want to keep going through the same interactive procedure on each system.

You can run the upgrade in unattended mode. For example, the following command upgrades a system from a software “depot” under the directory `/mydepot` on a network server named `petunia`:

```
/usr/sbin/upgrade -o -s petunia:/mydepot -x autoreboot=true
```

Caution

Run the upgrade unattended *only if*:

- You have already run `snoop` and eliminated all **PROBLEMS** (see Chapter 4.)
- You have already successfully upgraded at least one similar system from the same version of 9.x to 10.01.

Cluster: Cluster upgrades can be run unattended if these conditions are met, and the clients are booted to the server.

After the Upgrade

This chapter describes tasks you should do immediately or soon after you first upgrade from HP-UX 9.x to 10.01. It also provides guidance on some maintenance tasks that you may not need to do in the same timeframe as the upgrade, but which are related to it.

Crash	If your system crashed during the upgrade, or will not boot, follow directions on the next page for recovering your 9.x system.
HP Apps	Upgrade HP applications (and third-party applications distributed via SD) to 10.01 <i>before</i> upgrading the system to 10.10. See “Important: Order of Major Tasks”, a little later in this chapter, for more information.
9.07	If you have just upgraded a 9.07 system to 10.01, you must now load 10.01 applications and then upgrade to 10.10 immediately; the transitional 10.01 version you are now running is not supported for any other purpose. For now, ignore the list of “Tasks You Should Do Now” that follows and go right to “Loading New 10.01 Features and HP Applications” on page 7-9.

Recovery from a Crash or Catastrophic Failure

If your system crashed during the upgrade, or will not boot 10.01, recover your 9.x root disk from the backup you made with COPYUTIL (as described in “Backing Up Your System” in Chapter 4) and recover the other disks, if any, from the conventional backup tapes.

To recover the root disk:

1. Find the “Support Media” kit and the *Support Media User’s Guide* (HP part number 92453-90010).
2. Turn to Appendix A of the *Support Media User’s Guide*.
3. Put the tape or CDROM containing the offline diagnostics into the drive.
4. Boot the system from the tape or CDROM.

At the ISL prompt, enter:

```
ode copyutil
```

5. Follow directions in Appendix A of the *Support Media User’s Guide* to recover your root disk.
6. Repeat the above steps for any other disks backed up via COPYUTIL.

Now recover your data disks from the conventional backup.

When you are sure everything is working, and you have identified the cause of the original failure, you can start the upgrade again.

Important: Order of Major Tasks

You are probably reading this chapter right after upgrading your system from 9.x to 10.01. The most important tasks left to do are:

1. Make a recovery system.
2. Add new 10.01 features and applications not automatically chosen for you by **Match What Target Has**.
3. Upgrade to 10.10 (if you so decide).
4. Add 10.10 applications (if you are now on 10.10).

Caution

Note the order of the steps above.

You must upgrade HP applications to 10.01 *before* upgrading the system to 10.10. This allows the applications to go through the same kind of 9.x-to-10.x conversion HP-UX itself has just been through, moving application files to new locations, creating new device files, etc.

It will *not* work to upgrade an HP application directly from 9.x to 10.10, without installing the 10.01 version first.

See “Loading New 10.01 Features and HP Applications”, later in this chapter, for instructions on adding applications. See “HP Applications” in Chapter 4 for information on which HP applications are supported on 10.01.

A more detailed list of tasks starts on the next page.

Tasks You Should Do Now

Do these tasks immediately after you upgrade the system:

1. Set the `MANPATH` variable.
See “Setting and Using Pathnames”.
2. Analyze the upgrade logs and follow any directions you find in them.
See “Examining the Upgrade Log Files”.
3. Make a recovery system.
See “Making a Recovery System”.
4. Load and configure new HP-UX 10.01 features, and HP applications not bundled with HP-UX, if necessary.
See “Loading New 10.01 Features and HP Applications”.
5. Upgrade to 10.10, if you so decide.
See “Updating to 10.10”.
6. “Resecure” the system, if necessary.
See “Resecuring a Trusted System Cluster”.
7. Reorganize directories, if you so decide.
See “Reorganizing Your Directories”.
8. Add drivers that have been removed, if necessary.
See “Adding Drivers”.
9. Recreate device-file symbolic links and check for instances of `mknod`.
See “Checking Device Files”.
10. Modify startup and shutdown scripts.
See “Modifying System Startup and Shutdown Scripts”.
11. Modify users’ scripts if necessary.
See “Modifying User Scripts”.
12. Check the documentation for known problems.

7-4 After the Upgrade

See “Checking for Known Problems”

13. Test and certify applications if necessary.

See “Testing and Certifying Applications”.

14. Move converted code and scripts into place, if necessary.

See “Moving Code and Scripts into Place”.

Tasks You Can Do Later

The tasks in the following list do not need to be done immediately; you may not need to do some of them at all, but you should read through the sections now so that you can make informed decisions.

- Convert disks to LVM (Series 700)

See “Converting Disks to LVM (Series 700)”.

- Extend logical volumes.

See “Extending File Systems”.

- Remove software you don’t need.

See “Removing Software You Do Not Need”.

- Configure new NFS Diskless clusters (on systems that were *not* part of an HP-UX (“DUX”) cluster on 9.x).

See “Setting Up a New NFS Diskless Cluster”.

- Maintain transition links.

See “Maintaining Transition Links”.

- Convert code and scripts if necessary.

See “Converting Code and Scripts”

Setting and Using Pathnames

Setting MANPATH

Modify your `MANPATH` variable to include `/opt/upgrade/share/man`. This allows you to see the manpages for the analysis tools described in Chapter 5.

Using New 10.x Pathnames

The upgrade software has modified your `root` user `.profile` to include the 10.01 paths `/sbin` and `/usr/sbin` in the `PATH` variable. This means that you can use new 10.01 commands such as `swinstall` without specifying the full path.

You can also in most cases use familiar commands such as `shutdown` without specifying the pathname, or you can use the 9.x pathname. This is because **transition links** link the 9.x pathnames to the 10.x pathnames.

But this may not work if you are in single-user mode.

Using 9.x Pathnames in Single-User Mode

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

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

```
/etc/reboot
```

the system will respond:

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

The command you need in this case is:

```
/sbin/reboot
```

7-6 After the Upgrade

Checking Run-Levels

The run-level needed for many functions has increased by one as 10.0; for example, run-level 4 is needed to initiate VUE. See chapter 1 of the 10.01 version of the *HP-UX System Administration Tasks* manual, under “Controlling Usage and Processes with Run-Levels”, for more information.

If you have not already done so, check scripts and procedures for assumptions about run-levels that may not be valid on 10.x.

Examining the Upgrade Log Files

Examine the log files that the upgrade process has created and verify that all the software has been loaded as you expected, and that the process has run without any serious problems.

Note	Before you do anything else, copy <code>/etc/rc.log</code> to another name. This file contains valuable information (showing what <code>upgrade</code> did to convert system configuration scripts) and it will be over-written next time you boot the system.
-------------	--

You should examine the following files under `/var/adm/sw/`:

```
swagent.log
swagentd.log
swinstall.log
swpackage.log
upgrade.log
```

You can also run `swverify(1m)` to check that all the software has been installed correctly and all dependencies are met.

Note

After running `swverify`, you may see errors such as these in the logfile `/var/adm/sw/swagent.log`:

```
ERROR:   File "/tmp/dummy.xxx" missing.
ERROR:   Fileset "HPUX9.fileset,l=/,r=" had file errors.
...
ERROR:   Verify failed HPUX9.fileset,l=/,r=
```

You can ignore these **ERRORs**; they relate to dummy filesets that the upgrade tools created so that the **Match What Target Has** operation would work correctly. (See “What You Will See” in Chapter 4 for more information about **Match What Target Has**.)

LVM Conversion “Error” in a Cluster

You may find an **ERROR** in the `rc.log` file that is actually not an error. The error message looks like this:

```
ERROR:   Migration of SDS disk(s) to 10.x format unsuccessful.
Details follow:
```

[Details differ depending on the system configuration]

You may see this message in the server’s `rc.log` or a client’s `rc.log.old` (after running `upgdiskless`). Log in to the server or client as root and run

```
vgdisplay -v
```

- If `vgdisplay` shows that one volume group now exists for each SDS array that was configured into the 9.x system, you can ignore the error message in `rc.log`.
- If you do not see one volume group per SDS array, the error is real; call HP.

Making a Recovery System

You need to create a failsafe backup of the new 10.01 system. Do this with COPYUTIL, as described under “Backing Up Your System” in Chapter 4.

Caution Do this even if you plan to update the system to 10.10 right away; if the system should crash sometime between now and the end of the 10.10 update, and you had no 10.01 backup, you would have go back to 9.x and start over.

Loading New 10.01 Features and HP Applications

Loading New 10.01 Features

So far, you have loaded and booted a system selected for you by the **Match What Target Has** operation. This is a 10.01 system that is functionally equivalent to your 9.x system; that is, it has the HP-UX 10.0 file system layout and the new versions of HP-UX commands and libraries, but it does not have all of the new features that became available as of 10.0 and 10.01, such as the **Journalized File System (JFS)**

You may or may not want these new features.

- If you don't want any of the additional features, proceed to "Loading Applications".
- If you want some of the additional features, or all of them, follow the procedure on the next page.

You will be using `swinstall(1M)` (or `swcluster(1M)` for an NFS Diskless server); the `upgrade` tool is no longer on your system, but `swinstall` and `swcluster` will look very similar to what you've already seen (`upgrade` ran `swinstall` for you to load the software).

Procedure

1. If you have not already done so, read the discussion of “bundles” and “products” under “What Are Bundles and Products?” in Chapter 4.

Use that section, the discussion of new features in “I/O Convergence” and “Other Operating System and Subsystem Changes” in the *Release Notes for HP-UX 10.10*, and the table of bundles and products in Appendix A of this manual, to decide exactly what you are going to load.

2. If you are upgrading a small Series 800 system and you intend to load a “non-graphics” bundle, make sure you have followed all the directions for removing software in “Series 800 Small System” in Chapter 4.

If you didn’t follow these directions, the graphics and NLS filesets may still be on your system. You can remove them with `swremove`; see “Removing Software You Do Not Need”.

3. Check that the `swagentd` demon is running:

```
ps -ef | grep swagentd
```

If `swagentd` is not already running, start it:

```
/usr/sbin/swagentd
```

4. Run `swinstall`:

```
/usr/sbin/swinstall
```

Cluster On an NFS Diskless cluster server, run `swcluster` instead:

```
/usr/sbin/swcluster -i
```

You can also use SAM to update the cluster. See *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10* for more information.

5. Identify the source of the software to be loaded (tape, CD-ROM or network server) as you did during the initial upgrade (see “Running the Upgrade” in Chapter 6).

6. Choose the software you want to add.

- For example, you might choose `HPUXEngRT700`, the English-language run-time bundle for the Series 700.

This and the other run-time bundles include all the new 10.01 products and features, but they do not include the large filesets for Native Language Support in languages other than the language chosen.

- For a small Series 800 system, you may have decided to choose a “non-graphics” bundle; see “Removing Unneeded Filesets” in Chapter 4 for a discussion of this option.

Double-click on the bundle to examine its contents, as described in “Running snoop” in Chapter 4. See “What You Will See” in Chapter 4 for a discussion of products and bundles.

7. Pull down the **Actions** menu and choose **Mark for Install**.

8. Choose **Install (analysis)**.

9. If no errors are reported from the analysis phase, let the software installation proceed.

If there are errors, do not proceed. Exit the program and follow directions for this case in “Running the Upgrade” in Chapter 6.

10. If necessary, let SD reboot the system and log back in.

11. Review the log files again (see “Examining the Upgrade Log Files”, earlier in this chapter).

Note

When you review `/var/adm/swagent.log` you may notice a number of entries saying that filesets were “skipped”. This is ok: it means that the 10.01 version of the fileset was already on the system (that is, it was installed when you upgraded with **Match What Target Has**).

Loading Applications

Now you should load any HP-supplied 10.01 products that were not on the “CORE” tape(s) or CDs.

Use `swinstall` (or `swcluster`) to do this, as described in the previous subsection, but make sure you find and follow directions in the documentation that came with the products (these documents are often called *Release Notes for [x]*, where *[x]* is the name of a specific product).

Caution Not all HP applications are currently supported on 10.01. See “HP Applications” in Chapter 4 for a list of those that are, but note that even these are not guaranteed to work on NFS Diskless clusters and the other systems listed under “New Upgrade Capabilities” in Chapter 1.

Contact your HP Sales Representative (SR) for more information.

Updating to 10.10

If you have followed the procedures in this manual so far, the upgrade to 10.10 should be smooth and simple.

The procedures that follow are a “cookbook” for getting to 10.10; they should be sufficient if you feel comfortable using SD and intend to use the same hardware configuration to load 10.10 as you used to upgrade to 10.01 (for example a DDS drive attached to the system that is being updated). If you think you may need more help, find the manual *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*, which is packaged with the 10.10 software, and use the chapter “Updating from HP-UX 10.x to 10.10” instead.

Caution If you have not yet upgraded your HP applications to 10.01, *DO NOT PROCEED* with the upgrade to 10.10. Go back to “Loading New 10.01 Features and HP Applications”.

Updating SD

Before you can update to 10.10, you *must* extract the new version of Software Distributor (SD) from the 10.10 tape, CD or software depot from which you plan to update your system.

Caution **Do not use the 10.01 version of `swinstall` to update the system to 10.10.** The update will fail.

(`swcluster`, used to update an NFS Diskless server, will also fail; it calls `swinstall`.)

What You Are Going To Do

To update SD, you must first load the `swgettools` utility onto your system, then use `swgettools` to get the new version of SD.

Using swgettools to Update SD

Disk Space. The `swgettools` command needs a temporary directory with at least 11 Mb of free space. By default, `swgettools` will use the `/var/tmp` directory. If there is not enough space in the temporary directory, `swgettools` will fail.

You can tell `swgettools` to use a different temporary directory by means of the `-t dir_path` command-line option. Do this if you do not have 11 Mb free in `/var/tmp`; use `bdf /var/tmp` to check.

Loading swgettools. `swgettools` is shipped in the `catalog/SW-DIST/pfiles` directory. Depending on whether the 10.10 software is on CD, tape or a remote system in a software depot, use `cp`, `tar`, or `rcp` respectively to load `swgettools` onto your system.

For example, to load `swgettools` from a CD-ROM mounted at `/mnt/cdrom` into `/var/tmp`:

```
cp /mnt/cdrom/catalog/SW-DIST/pfiles/swgettools /var/tmp
```

See “Examples” below for more examples and other options.

Getting the New SD Tools. Now use `swgettools` to update SD; for example:

```
/var/tmp/swgettools -s /mnt/cdrom
```

where `-s /mnt/cdrom` indicates a CD-ROM drive mounted on `/mnt/cdrom`.

You can now update your system to 10.10. Follow directions in the next subsection, “Running the Update”.

See “Examples” below for more examples.

Caution **Do not reboot your system** after running `swgettools` and before you run `swinstall` or `swcluster` to update HP-UX.

If you do have to reboot, you must run `swgettools` again before updating HP-UX.

Examples. To install the new SW-DIST product from CD-ROM at `/mnt/cdrom_depot`:

```
cp /mnt/cdrom_depot/catalog/SW-DIST/pfiles/swgettools /var/tmp
```

```
/var/tmp/swgettools -s /mnt/cdrom_depot
```

To install the new SW-DIST product from tape at `/dev/rmt/0m`:

```
cd /var/tmp
tar -xvf /dev/rmt/0m catalog/SW-DIST/pfiles/swgettools
cp /var/tmp/catalog/SW-DIST/pfiles/swgettools /var/tmp/swgettools
rm -rf /var/tmp/catalog
/var/tmp/swgettools -s /dev/rmt/0m
```

To install the new SW-DIST product from a remote depot on system `swperf` at `/var/spool/sw`:

```
rcp swperf:/var/spool/sw/catalog/SW-DIST/pfiles/swgettools /var/tmp
/var/tmp/swgettools -s swperf:/var/spool/sw
```

Updating SD without Root Access. If your users will be running `swgettools` against a depot on a remote server, and you do not want to grant them `rcp` access as root, follow one of the two procedures below (**Option 1** or **Option 2**).

Option 1

1. Copy the `swgettools` script file and the `swlib.tar.Z` file from the tape or CD (in the `catalog/SW-GETTOOLS/pfiles` directory) to a location that your users have FTP access to.
2. Tell the user to:
 - a. FTP the two files into the `/var/tmp` directory on the system to be updated.
 - b. Use `chmod +x` to make the `swgettools` script executable.
 - c. Run `swgettools` and specify the remote depot with the `-s` option (and, if necessary, `-t` to specify a temporary directory other than `/var/temp`).

Option 2

Tell your users to:

1. Copy the new SW-GETTOOLS product from the remote depot to a local depot.

For example, to copy SW-GETTOOLS from a remote CD-ROM depot registered at `swperf:/var/spool/sw` to a local depot in `/tmp/depot`:

```
swcopy -s swperf:/mnt/cdrom SW-GETTOOLS @ /tmp/depot
```

2. Copy the `swgettools` script file and the `swlib.tar.Z` file to the `/var/tmp` directory; for example:

```
cp /tmp/depot/catalog/SW-GETTOOLS/pfiles/sw* /var/tmp
```

3. Run `swgettools` and specify the remote depot with the `-s` option, for example:

```
/var/tmp/swgettools -s swperf:/mnt/cdrom
```

Note

If you will be using a temporary directory other than `/var/tmp`, copy the `swgettools` script and the `swlib.tar.Z` file to the temporary directory you will be using and use the `-t` option on the `swgettools` command line to specify its location.

For example:

```
cp /tmp/depot/catalog/SW-GETTOOLS/pfiles/sw* /usr/tmp  
/usr/tmp/swgettools -s swperf:/mnt/cdrom -t /usr/tmp
```

Running the Update

Caution

Do not try to update to 10.10 without first updating SD itself; see the previous subsection, “Updating SD”.

Once you have updated SD, follow the procedure that starts on the next page to update your system to 10.10.

1. Log in as superuser.
2. Run `swinstall`:

```
/usr/sbin/swinstall
```

Cluster On an NFS Diskless cluster server, run `swcluster` instead:

```
/usr/sbin/swcluster -i
```

You can also use SAM to update the cluster. See *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10* for more information.

3. Identify the source of the software to be loaded (tape, CD-ROM or network server) as you did during the upgrade to 10.01 (see “Running the Upgrade” in Chapter 6).
4. Choose **Match What Target Has** from the **Actions** menu.
5. Select any additional features of 10.10 you want, pull down the **Actions** menu again and choose **Mark for Install**.

For example, you may want to add the HP Common Desktop Runtime Environment (CDE 1.0), the new industry standard UNIX desktop which 10.10 is offering in addition to, and as a successor to, VUE. See the *Release Notes for HP-UX 10.10* for details.

6. Choose **Install (analysis)**.
7. If no errors are reported from the analysis phase, let the software installation proceed.

If there are errors, do not proceed. Exit the program and follow directions for this case in “Running the Upgrade” in Chapter 6.

8. Let SD reboot the system.

Cluster Once the server is up and running, reboot the clients.

9. Log back in.

10. Review the log files again (see “Examining the Upgrade Log Files”, earlier in this chapter).

Loading 10.10 Applications

Now you may need to run `swinstall` again to load any HP-supplied 10.10 products that were not on the “CORE” tape(s) or CDs.

Cluster ■ Use `swcluster -i` on the server to install applications in an NFS Diskless cluster.

 ■ On a 10.10 system, running the `shutdown` command on the server shuts down NFS Diskless clients as well (this was not true on 10.01).

Make sure you read the product installation instructions (often called *Release Notes for [product-name]*) first.

Caution Not all HP applications are currently supported on 10.10. Contact your HP Sales Representative (SR) for more information.

Resecuring a Trusted System Cluster

This section applies only to an NFS Diskless server that under 9.x was:

- An HP-UX (“DUX”) cluster server; *and*
- A Trusted System

To “resecure” a server that you “unsecured” in order to upgrade it, follow directions in chapter 12 of the 10.01 version of the *HP-UX System Administration Tasks* manual. Find the section of that chapter called “Configuring NFS Diskless Clusters for Trusted Systems”; it begins on page 12-33. Follow the procedure that corresponds to the “policies” you set in `upgdiskless`; for example, if you accepted the SHARED default, use the procedure on page 12-34, “Choice 2: Clusters with Shared Password Databases”.

Reorganizing Your Directories

HP “core” commands, and applications and utilities bundled with the operating system, have been moved to new locations during the upgrade to comply with the HP-UX 10.0 filesystem layout described in the *HP-UX 10.0 File System Layout* white paper. “Non-core” HP applications you have loaded onto a 10.01 system (following directions in previous sections of this chapter), will also have been reorganized to conform to the 10.0 filesystem-layout guidelines.

Your own user and application directories have *not* been moved during the upgrade to 10.01 (unless they were subdirectories of HP-UX 9.x directories such as `/bin`; see “Preparing for File System Layout Changes” in Chapter 4 for details).

Whether you reorganize these directories yourself is up to you. You should at least read the *HP-UX 10.0 File System Layout* white paper and understand the rationale for the new layout; and for NFS Diskless clusters you need to read the *NFS Diskless Concepts and Administration* white paper as well and follow its recommendations.

Adding Drivers

The upgrade process has removed non-HP drivers, and HP drivers in filesets that you did not load. The drivers have been removed from the kernel, but the original `dfile` or `gen` file that includes them has been saved as `/stand/system.old`. The original master file has been saved as `/etc/upgrade/save/hostname/etc/master`. Upgrade has also removed the device files associated with these drivers.

You may still be able to use these drivers. For information on adding non-“core” and custom drivers, see Appendix C, “Major and Minor Numbers,” in the 10.0 version of *Configuring HP-UX for Peripherals*.

Checking Device Files

Most of the device files on your 9.x system have been replaced during the upgrade to 10.01, but their names have been preserved as symbolic links to the new files. This means that code and scripts that invoke the old names will continue to work on 10.x.

Check for:

1. Use of `mknod(1M or 2)`, in code, scripts, or written procedures to create device files.

Use `mksf` or `insf` instead of `mknod` in 10.x. For more information see “I/O Convergence” under “Major Changes for HP-UX 10.0 and 10.01” in the *Release Notes for HP-UX 10.10*, and the 10.0 version of the manual *Configuring HP-UX for Peripherals*.

If you have not already done so, use the `prepare(update_aid)` tool to check for calls to `mknod`; see Chapter 5 for details.

2. Symbolic links to 9.x device-file names.

You were advised in Chapter 4 to make a note of any such links that existed on your 9.x system. Use this list to recreate the links now.

Modifying System Startup and Shutdown Scripts

During upgrade, your 9.x scripts in `/etc/shutdown.d` are moved to `/etc/#shutdown.d`. You need to modify these scripts so that they will work with the new 10.0 shutdown method.

Do the following:

1. Select the run level at which the scripts should be executed when transitioning from higher run levels. In the 10.0 scheme, shutdown scripts are executed whenever `init(1M)` is executed to bring the system to a lower-numbered run level, as well as when `shutdown(1M)` is executed.

For example, if the shutdown script shuts down a subsystem that is started in run level 2, then the corresponding shutdown script would probably be executed when entering run level 1 from a higher run level.

Note that some 10.x run levels have been increased by one from the corresponding 9.x run levels. (For example, 10.x HP VUE requires run level 4, and the default non-VUE run level is 3.)

2. Select the sequence numbers for the startup and shutdown scripts; the sequence number determines the order in which scripts are executed *within the selected run level*.
3. Modify the 9.x scripts so that they follow the same format as the template in `/sbin/init.d/template` (following the directions contained in the comments in the template), and place the scripts in the `/sbin/init.d` directory.
4. Create symbolic links to these scripts as follows:
 - a. Place startup script links in the `/sbin/rcN.d` directory, (where N is the run level in which you want the subsystem to be started), and place shutdown script links in the `/sbin/rcN-1.d` directory (so that the subsystem will be shut down when entering run level $N-1$ from a higher run level).

The naming convention for startup and shutdown links is as follows:

```
startup:    S<seqn><script_name>
shutdown:  K<seqn><script_name>
```

where **S** stands for “start” and **K** stands for “kill”, **seqn** is the sequence number, and **script_name** is the name of the script in `/sbin/init.d`. Sequence numbers for startup and shutdown are typically not the same.

For more information on the 10.0 system startup/shutdown model, see the *HP-UX 10.0 File System Layout* white paper.

Note

`upgdiskless` leaves configuration files in the “save area”, `/etc/upgrade/save/name`, where *name* is the official hostname of your system as returned by `uname -a`.

For clients of clusters that have been converted from 9.x “DUX”, look under `/etc/upgrade/save/client` on the client or `export/shared_roots/client/etc/upgrade/save/client` on the server. If `upgdiskless` did not fully convert the client (because it was not booted to the server during upgrade or for some other reason) look under `/etc/upgrade/save/client` on the server. (If you chose the default **SHARED** password-file policy in `upgdiskless`, the files are also available on the client under `/net/server/etc/upgrade/save/client`.)

Modifying `/etc/inittab`

If you added your own custom entries to `/etc/inittab` on the 9.x system, and you still want to use them under 10.x, you need to merge them in by hand; `upgrade` has not copied them into the new `/etc/inittab`, but has preserved them in the file

```
/etc/upgrade/save/hostname/tttools/OS.Core.d/cmds/inittab.mods
```

Cluster

In a cluster, custom entries have been preserved in separate files for the server and the clients. The above path contains the server’s entries if you’re logged in to the server, or the client’s if you’re logged in to a client.

You will probably need to change run-levels in these custom entries; run-levels for most functions have increased by one as of 10.0.

Modifying User Scripts

Scripts that use commands or options that have changed or are no longer supported, or invoke `mknod` to create device files, need to be modified to run on 10.x.

See the sections on I/O convergence, shells, and commands and libraries in the *Release Notes for HP-UX 10.10* for information on the changes.

If you have not already done so, run `prepare(update_aid)` to check and optionally modify 9.x user scripts. Don't forget users' `.login` and `.profile` scripts. See Chapter 5, and “Operating 9.x and 10.x Systems Together” in Chapter 2, for more information.

Note `prepare` and the other tools described in Chapter 5 have been moved during the upgrade from `/upgrade/bin` to `/opt/upgrade/bin`.

HP VUE Configuration files

HP-UX 10.01 HP VUE configuration files have moved from where they were on 9.x. The upgrade process does not automatically convert the 9.x HP VUE configuration files.

The following configuration files are affected:

under /usr/vue/config/:

- Xaccess
- Xconfig
- Xfailsafe
- Xreset
- Xresources
- Xservers
- Xsession
- Xstartup
- sys.font
- sys.res.lite
- sys.resources
- sys.res.lite
- sys.session
- sys.vueprofile
- sys.vuemrc
- vuerc

under /usr/vue/config/panels/

- fp.clock
- fp.date
- fp.help
- fp.home
- fp.load
- fp.mail
- fp.printer
- fp.style
- fp.terminal
- fp.tool
- fp.toolbox
- fp.trash

under /\$HOME/.vue/

- config/vuemrc
- panels/fp.*
- types/*.vf

To apply these configuration files to your 10.x system, do the following:

7-26 After the Upgrade

1. If necessary, change pathnames within the file to HP-UX 10.x pathnames.
2. Reapply the file after you have started an HP-UX 10.x session.

Refer to *Using Your HP Workstation* or the *HP Visual User Interface 3.0 User's Guide* for detailed information.

X Window System Configuration Files

The following X Windows configuration files are moved to `/etc/X11` during the upgrade:

```
XOscreens
XOdevices
XOpointers
rgb.txt
```

These files are not changed but each file in the old path, `/usr/lib/X11`, has a link pointing to the new path. You can use either path, but the new path is preferable.

To apply these configuration files to your HP-UX 10.x system, do the following:

1. If necessary, change pathnames within the file to HP-UX 10.x pathnames.
2. Reapply the file after you have started an HP-UX 10.x session.

Refer to *Using the X Window System* if you need directions.

The `XOscreens` file has a new format, but the server can read both the old and new formats. Though it is not necessary to convert to the new format for 10.x, you should do so at some convenient time; the old format may not be supported in future releases. The default HP-UX 10.x-format `XOscreens` file is under `/usr/newconfig/X11`.

HP MPower

HP MPower cannot be upgraded to, and does not run on, HP-UX 10.01.

A separate product providing complete MPower 2.0 functionality on HP-UX 10.x systems will be available in future.

Troubleshooting

(1)PEX. Your system may need a symbolic link that it does not have. If you are experiencing problems, check for this link under `/usr/lib/extensions`:

```
libXhpPEX.sl -> /usr/lib/X11/extensions/libXhpPEX.1
```

If the link does not exist, do the following (you must be the root user):

```
cp /opt/graphics/PEX5/lib/libXhpPEX.1 /usr/lib/X11/extensions/libXhpPEX.1
ln -s /usr/lib/X11/extensions/libXhpPEX.1 /usr/lib/X11/extensions/libXhpPEX.sl
```

(2)After Removing Softbench Products. If you install and then remove HP Softbench products on a 10.01 system, users may find they cannot start VUE. This would happen if the `/opt/softbench` directory could not be cleaned up properly because some of the files were busy when you removed the product.

Symptom

Users will see an error when they try to start VUE, telling them VUE cannot be started and advising them to check `/etc/rc.config.d/netconf`, `/etc/hosts` and `/var/adm/inetd.sec`.

This symptom is usually caused by a configuration error in one of these files; but it could also be a result of the cleanup failure described above.

Solution

To solve the problem, or to determine if the `/opt/softbench` directory is causing the problem, move the directory; for example,

```
mv /opt/softbench /opt/softbench.old
```

Then try restarting VUE.

Checking for Known Problems

Check the “Major Changes for HP-UX 10.0 and 10.01” section of the *Release Notes for HP-UX 10.10*, under “Known Problems and Workarounds” for known problems in HP-UX 10.01.

The ASCII file for *Release Notes* is on your system in the directory `/usr/share/doc/10.10RelNotes`. VUE users can also use the hypertext version: click on the documentation icon (books and a question mark) on the dashboard.

Moving Code and Scripts into Place

If you have used the `prepare` and `analyzer` tools to create 10.x-compliant versions of code and scripts, you now need to move these files into their “production” locations.

If you have reorganized your application and user directories along the lines recommended in the *HP-UX 10.0 File System Layout* white paper, don’t forget that these “production” locations are probably not where they were on 9.x.

Testing and Certifying Applications

Before putting your system back into production, you will need to run the regression and certification tests you normally run when upgrading to a major new release of HP-UX.

You need to do this even if you have already performed the analysis and conversion tasks described in Chapter 5.

Ideally, you should already have done this on a test machine. If so, now is a good time to bring the applications over from the test machine.

Converting Disks to LVM (Series 700)

Should You Convert to LVM?

Consider the following factors:

1. How many disks do you have on this system?
2. Do you need the flexibility that LVM offers?
3. What is involved in administering LVM?

How Many Disks Do You Have on this System?

If your system has only one or two disks, and you are satisfied with the current distribution of data, then you may not need to move to LVM, but you can do so at any time.

Do You Need the Flexibility of LVM?

LVM makes it easy for you to expand or reduce the disk space allocated for a given purpose (such as a file system or swap area) and provides the means to spread a filesystem over multiple physical disks.

Read the sections “What Are Logical Volumes?” and “Who Should Use Logical Volumes?” in chapter 3 of the 10.01 version of the *HP-UX System Administration Tasks* manual for a brief introduction to the capabilities and benefits of LVM.

What Is Involved in Administering LVM?

LVM is easy to learn because you can do most tasks via SAM, the System Administration Manager tool. The *HP-UX System Administration Tasks* manual provides step-by-step instructions for tasks for which you can *not* use SAM.

Summary

HP recommends LVM for all but the most simple configurations. Read chapter 3 of the *HP-UX System Administration Tasks* manual to learn more about LVM and decide whether or not your system will benefit.

Note Remember that if you had SDS disk arrays on your 9.x system, they have been converted to logical volumes during the upgrade, and single SDS disks are being managed by the compatibility pseudo-driver (`cpd`), as explained in “Considerations for Series 700s” in Chapter 4.

Converting the Root Disk to LVM

If you decide to convert your root disk to LVM, you will need to “re-install” the version of HP-UX you are currently running (10.01 or 10.10). This process erases your root disk, reconfigures it for LVM, and reloads HP-UX onto the disk.

9.07 **Caution:** If you have upgraded from 9.07, you *cannot* re-install 10.01. If you want to convert your root disk to LVM, upgrade to 10.10 first, then re-install 10.10.

Follow directions near the end of chapter 3 of the 10.01 version of the *HP-UX System Administration Tasks* manual, under “Converting a Root Disk”. You will also need the manual *Installing HP-UX 10.10 and Updating from HP-UX 10.x to 10.10*.

Note You can avoid re-installing HP-UX if you have a spare disk to which you can move the contents of your current root disk. See “Converting Your Current Root Disk by Using a Spare Disk” in chapter 3 of the *HP-UX System Administration Tasks* manual.

Removing LVM

If you were running SDS on 9.x, the upgrade process automatically installed LVM on the system. If you were using SDS disk arrays, they are now being managed by LVM; otherwise, if you have decided not to use LVM for now, you can remove it from your system:

```
/usr/sbin/swremove LVM.LVM-RUN
```

or (in a cluster):

```
/usr/sbin/swcluster -r
```

Note

If you were using SDS, but only for single-disk striping, these disks are *not* being managed by LVM. You can safely remove LVM from your system, but remember that if a former SDS disk is damaged or crashes, you will need LVM if you want to continue to manage the disk's data in "stripes". See "Considerations for Series 700s" in Chapter 4.

Extending File Systems

Extending Logical Volumes

You may need to extend the logical volumes for `/var`, `/usr` and `/opt`.

You cannot use SAM to do this because SAM uses these directories, and so cannot unmount them; you may also have trouble unmounting them yourself because HP-UX daemons have open logfiles there.

The best way to extend volumes containing these directories is this:

1. Reboot the system.
2. Interrupt the boot to interact with ISL.
3. Specify that the system is boot into `init` state `S` (single-user mode, in which no volumes will be mounted):

```
ISL> hpux -iS boot
```

4. Then use the `lvextend` and `extendfs` commands, as described in the *HP-UX System Administration Tasks* manual.

Extending a Journaled File System

- If you have the optional product *HP OnlineJFS*, you can extend a Journaled File System (VxFS) provided the logical volume on which it was created has enough space, for example:

```
lvextend -L newsize logical_vol  
fsadm -b newFSsize raw_volume_for_this_file_system
```

- If you have *not* purchased *HP OnlineJFS*, and need to extend a journaled file system, contact your HP Representative or see HP SupportLine for more information.

Removing Software You Do Not Need

Removing 10.x Software

Use `swremove(1M)` (or `swcluster` on a cluster server) to remove software from a 10.x system,

```
/usr/sbin/swremove
```

or (in a cluster):

```
/usr/sbin/swcluster -r
```

`swremove` looks and behaves very much like `swinstall`, which you ran under the `upgrade` program, as described in “Running the Upgrade” in Chapter 6. But in this case, when you “mark” software, you are selecting it for removal from, rather than installation onto, your system.

`swremove` checks dependencies; that is, it prevents you from removing software that other software you are not removing requires. But you still need to be careful to remove only functionality no user of your system, or application running on the system, uses.

Note

The Confirmation dialog box in (in which you tell `swremove` whether or not you want to go ahead and remove the software you’ve chosen) refers to a **Disk Space** button on 10.01. Ignore this part of the message: `swremove` does not do a disk space analysis; but it can show you the size of each product or fileset you may want to remove.

To see how much disk space you save by removing software, do a `bdf` before and after running `swremove`.

You can run `swremove` in “unattended mode” if you know exactly what you want to remove; see the manpage for more information,

If you are not acutely short of disk space, it may be better to wait for a while before cleaning up the system. At that point, you may want to use the

`freedisk(1M)` tool, described under “Small Disks and Memory” in chapter 4, “Other Operating System and Subsystem Changes”, of the *Release Notes for HP-UX 10.10*.

`Freedisk` is another way of removing files; it invokes `swremove` to do the job of removing them, but its advantage over using `swremove` alone is that `freedisk` tells you what files you have not used.

See the manpage for more information.

Removing 9.x Software

In a few cases, the `upgrade` process leaves files in the “save area” (`/etc/upgrade/save/hostname`) that you’re not likely to need; for example

```
/etc/upgrade/save/hostname/usr/sam/config/rt/hostname.ascii  
/etc/upgrade/save/hostname/usr/sam/config/rt/hostname.noascii
```

Check these files to make sure you don’t need them (you may want to save log files, for example) and remove any you have no use for.

Cluster To check for context-dependent files (CDFs) left over from 9.x (“DUX”) clusters, use the following command:

```
find path-name-list -perm -04000 -type d -print
```

For more information, see:

- “Recommendation for Routine Backups on Existing 9.x Systems” in Chapter 4.
 - “Modifying HP-UX Clusters” in Chapter 4.
-

Setting Up a New NFS Diskless Cluster

Other sections in this manual have explained how to manage the conversion from 9.x “DUX” to 10.x NFS Diskless, but you may also have decided to configure some newly-upgraded standalone systems into an NFS Diskless cluster (see “Sharing Disk Space” in Chapter 2).

See chapter 11 of the 10.01 version of the *HP-UX System Administration Tasks* manual for information on configuring a new NFS Diskless server and adding clients.

Maintaining Transition Links

You should *not* normally need to install transition links.

They are installed by default during the upgrade to 10.01, and any time you add 10.01 software using SD commands.

If you suspect that someone has removed the transition links from your system, you can check by entering a command such as:

```
ll -R / | grep '^1.....[tT]'
```

Output should include lines such as:

```
lr-xr-xr-t 1 root sys 8 Jan 12 19:39 bin -> /usr/bin
lr-xr-xr-t 1 root sys 8 Jan 12 19:39 lib -> /usr/lib
```

Note The “l” characters in the above example are the letter “l” (ell), not the numeral 1 (one).

If you don’t see these links, it could be because someone has removed them. You can install them using `tlinstall(update_aid)`.

It is also possible, if someone other than you performed the upgrade, that they chose not to load the transition links fileset (even though it is one of the “core” filesets which should be loaded on all systems) or that someone has removed it.

You can check to see if the fileset is on your system by entering the command,

```
swlist -l fileset Upgrade.UPG-TLINK
```

You should see,

```
Upgrade.UPG-TLINK      B.10.nn      Upgrade tlink
```

- where *nn* is 01 or 10 depending on which version of 10.x you are running.

If the fileset is not on your system, use `swinstall(1M)` (or `swcluster(1M)` for a cluster) to load it from the install/upgrade tape or disk (or other source); see the `swinstall(1M)` and `swcluster(1M)` manpages for more information on these commands. By installing the UPG-TLINK fileset, you will automatically create the links (`swinstall` runs `tlink`).

Converting Code and Scripts

If you have opted for the “Quick Start” approach outlined under “Option A: Quick Start” in Chapter 3, you will need to convert your code and scripts sometime over the lifetime of 10.x; HP does not intend to support transition links beyond 10.x.

Now is a good time to start sizing and planning this task; the `prepare` tool can help you do that; see “Analyzing Files” in Chapter 5 and the `prepare(update_aid)` manpage.

`fnlookup(update_aid)` will help you map 9.x pathnames to their 10.x equivalents.

The `update_aid` tools were moved to `/opt/upgrade/bin` during the upgrade to 10.01.

8

Compatibility between 9.x Releases and 10.01

This chapter summarizes 10.0/10.01 changes and requirements you particularly need to take note of before upgrading to 10.01, and before moving files from a 9.x to a 10.x system, and indicates sources of help and further information.

Applications

Code compiled on 9.x systems, and not re-linked or recompiled on 10.x, will run on 10.x without problems, unless it does one of the following:

- Invokes hard-coded pathnames that no longer exist as of 10.0 (see the section on the HP-UX 10.0 filesystem in the *Release Notes*).
- Uses `mknod(1M)` to create device files with 9.x names or major and minor numbers (see the section on I/O convergence in the *Release Notes*).

(But system default device-file names such as `/dev/lp` are still supported on 10.x.)

- Invokes HP-UX commands or command options that are no longer supported as of 10.0 or whose meaning or behavior has changed (see the sections on shells and commands in the *Release Notes*).
- Invokes HP-proprietary library routines for HP-UX “diskless” clusters (see “HP-UX (“DUX”) Clusters No Longer Supported” later in this chapter).
- Invokes HP-proprietary NLS (Native Language Support) routines that are no longer supported as of 10.0 (see “NLS Routines No Longer Supported” later in this chapter).
- Invokes certain `libc` routines on a system on which **transition links** have been turned off (see the section on libraries in the *Release Notes*).

The links are in effect by default on all 10.x systems (see “Upgrade Tools” in Chapter 3).

- Mixes `libc` pattern-matching routines in unusual ways (see “Pattern-Matching Routines Mixed in Unusual Ways” later in this chapter).
- Uses shadow password routines or the shadow password file (see “Shadow Password Routines” later in this chapter).
- Uses NetIPC routines that are no longer supported (see “NetIPC Routines No Longer Supported” later in this chapter).
- Uses certain routines that have been changed for reasons of standards conformance or I/O convergence (see “Other Changes Affecting Applications” later in this chapter.)

8-2 Compatibility between 9.x Releases and 10.01

Run the `prepare(update_aid)` tool to check application source code for these conditions and, optionally, to create copies with the correct pathnames for 10.x. See the previous chapter for instructions.

The following potential problems *will not* be caught by `prepare` and `analyzer`; they are described in more detail later in this chapter.

- An application may behave differently if it depends on 9.x math library error handling.

See “Changes in Error Handling for Math Library Functions” later in this chapter.

- Programs that run correctly on 9.x could encounter problems on 10.x because of a 9.0 linker bug.

The software included in the package that accompanies this manual (the “Upgrade Preparation Media” package) includes tools to check for the existence of such problems and fix them; see “Linker Defects” later in this chapter for details.

- Applications will get different behavior on floating-point convert-to-integer overflow.

The 9.x behavior was incorrect, and inconsistent between the Series 700 and 800. It has been corrected; see “Floating-Point to Integer Conversion” later in this chapter.

The sections that follow provide some general guidelines for portability between HP-UX releases, and specific guidance for some of the cases in the list above.

Guidelines for Writing Portable Applications

To make sure an application will port as easily as possible from one HP-UX release to another (not just from 9.x to 10.x), a programmer should:

- Avoid using archive `libc` with other shared libraries (even `libdld.sl`).

Use archived `libc` only if the entire application is linked archived.

(“Archived” means bound at link time; “shared” means bound at run time).

Compatibility problems could arise when `libc` interfaces are resolved out of both shared and archived `libc`. In the case of `libdld.sl`, the compatibility problems could arise when a program tries to load a shared library that has dependencies on `libc.sl`.

- Avoid using both archived and shared `libc` or making `libc` an explicit dependency of other shared libraries.

Compatibility problems could arise when `libc` calls are resolved out of both shared and archived `libc`, either directly or via a dependent shared library.

The general rule is: *always link with the shared version of `libc` whenever any other shared libraries are being used.*

- Avoid linking applications with the `-B immediate` option; and avoid using `BIND_IMMEDIATE` when loading `libc` with `shl_load(3X)`.

The reasons for this are:

1. Using these options forces all symbols to be bound at startup time, even symbols that are not required by the application. If one of the symbols is not in a future release, the application will no longer load.
2. `libc` was versioned in the past and may be again in the future. If an older version of a symbol depends on other symbols which are not present, then the application will not load.

8-4 Compatibility between 9.x Releases and 10.01

Specific Guidelines for Moving Code from 9.x to 10.x

This section describes conditions for which programmers should check their code, and change the code if they exist, before moving the application from a 9.x to 10.x system.

Pattern-Matching Routines Mixed in Unusual Ways

This section outlines the problems that could arise if your code uses:

- Combinations with 10.x `glob(3C)` or 10.x `regcomp(3C)`
- 10.x `wordexp(3C)` and pre-10.0 `wordfree(3C)`

These combinations are not likely to occur. But if they did, it would be when some 10.x code (the “allocation” routine) is dynamically loaded—via `shl_findsym(3X)` for example—and some pre-10.0 code from `libc` (the “execution” or “deallocation” routine) is linked archived into an application or dependent library.

Combinations with 10.x `glob(3C)` or 10.x `regcomp(3C)`. The combinations in question are:

- 10.x `glob(3C)` and pre-10.0 `globfree(3C)`.
- 10.x `regcomp(3C)` and pre-10.0 `regex(3C)` or pre-10.0 `regfree(3C)`.

Direct call

If the application code calling `glob(3C)` and `globfree(3C)` is compiled and linked in this manner on a pre-10.0 system—

```
cc -Wl,+e, __dld_loc -Wl,+e,errno -Wl,+e,_end -Wl,+e,malloc -Wl, \
-a,archive main.c -l:libdld.sl
```

—and the application is then run on 10.x, it will receive a SIGSEGV signal while executing the `globfree(3C)` code:

```

0x00003514  _globfre+001c  OR      1,0,4
0x00003518  _globfre+0020  COMIBT,= 0,26,_globfree+0048
0x0000351c  _globfre+0024  STW     26,12(0,1)
0x00003520  _globfre+0028  ADDIL   0x0,27
0x00003524  _globfre+002c  OR      1,0,5
0x00003528  _globfre+0030  LDWS    0(0,26),31
0x0000352c  _globfre+0034  BL      free,2
0x00003530  _globfre+0038  STW     31,16(0,5)
0x00003534  _globfre+003c  LDW     16(0,5),26
0x00003538  _globfre+0040  COMIBF,= 0,26,_globfree+0030
File: unknown Procedure: _globfree + 0x00000030 Line: unknown

```

```

segmentation violation (no ignore) at 0x00003528
(file unknown): _globfree +0x00000030: (line unknown)
>t
 0 _globfree + 0x00000030 (0, 0, 0, 0)
...

```

Similarly, an application combining 10.x `regcomp(3C)` with pre-10.0 `regex(3C)` or `regfree(3C)` (in the manner described for `globfree` above) will receive a SIGSEGV signal while executing `regex` or `regfree` code.

Unless a SIGSEGV signal handler is installed, the system will generate a core dump and terminate the program.

Remedy: Recompile and relink the application on 10.x, thus eliminating all pre-10.0 modules.

Indirect call from shared library

If the application code calls `glob(3C)` indirectly from a dependent shared library that is relinked on 10.x and calls `globfree(3C)` directly, then the application will get a load error and abort.

For example, if the application is compiled and linked in this manner on a pre-10.0 system—

```
cc -Wl,-a,archive -l:libdld.sl main.c -L. -l:libfoo.sl
```

—and the library is then recompiled and linked in this manner on a 10.x system—

8-6 Compatibility between 9.x Releases and 10.01

```
cc -c +z foo.c
ld -b foo.o -o libfoo.sl
```

—then the application will get a load error and abort when it is run on 10.x; for example:

```
/usr/lib/dld.sl: Unresolved symbol: _end (data) from /lib/libc.sl
/usr/lib/dld.sl: Unresolved module for symbol: _curbrk (data) from /lib/libc.sl
/usr/lib/dld.sl: Unresolved module for symbol: malloc (code) from /lib/libc.sl
Abort(coredump)
```

Similarly, the application will get a load error and abort if it combines 10.x `regcomp(3C)` and pre-10.0 `regex(3C)` or `regfree(3C)` in the manner just described.

Remedy: Recompile and relink the application on 10.x, thus eliminating all pre-10.0 modules.

10.x `wordexp(3C)` and pre-10.0 `wordfree(3C)`. Combining `wordexp(3C)` and `wordfree(3C)` directly, in a manner similar to that shown for `glob` and `globfree` above, will *not* cause a SIGSEGV signal, but will result in the pre-10.0 `wordfree` behavior: a memory leak of up to `strlen+16` bytes for each string in the `we_wordv[]` list. (See the `wordexp` man page on a 9.x or 10.x system for a description of `we_wordv[]`.)

Combining `wordexp` and `wordfree` indirectly, in a manner similar to that shown for `glob` and `globfree` above, will result in the same type of behavior as in the example under “Indirect Call from Shared Library”: unresolved symbols at load time.

Remedy: Recompile and relink the application on 10.x, thus eliminating all pre-10.0 modules.

Shadow Password Routines

The shadow password routines listed in the table below have been moved to a separate library for 10.x: `libsec`.

In addition, the standard password routines `fgetpwent(3C)`, `getpwaid(3C)`, `getpwent(3C)`, `getpwnam(3C)`, `getpwuid(3C)` and their re-entrant counterparts no longer call the corresponding shadow password routines automatically in 10.x.

Results of calling password routines from 9.x executable on 10.x. The shadow password routines have been stubbed out in the 10.x `libc`; the functional routines are in `libsec`. The shadow password file `/.secure/etc/passwd` does not exist in 10.x. See the table on the next page for the effects on 9.x code.

The table that follows shows the results a 9.x executable that calls the password routines will get on 10.x. Results depend on the type of system, how the library code was linked and which routines are called.

	On 10.x Trusted System	On 10.x Regular System
executable calling:	code linked shared archived	code linked shared archived
setspwent		return void;
setspwent_r		errno set to ENOENT
endspwent	error	
endspwent_r		return void; errno unchanged
fgetspwent		
fgetspwent_r	and	
getspwaid		return NULL;
getspwaid_r	abort;	
getspwent		errno set to ENOENT;
getspwent_r	see	
getspwnam		see Notes (3) and (4)
getspwnam_r	Note (1)	
getspwuid		
getspwuid_r		
putspwent		returns -1; errno set to ENOENT; see Note (4)
standard password routines (setpwent, etc.)	set password field to "*" and audit fields to -1; see Note (2)	ok; no change from 9.x

See remedies and notes on the next page.

- *Remedy if calling `fgetspwent(3C)` and `putspwent(3C)`: see Note 4.*
- *Remedy for all other cases: recompile and relink the application on 10.x, naming the security library explicitly in the compile or link statement(for example: `cc -o main main.c -lsec`).*

NOTES

Note 1

The stubbed shadow password routines in `libc` will not access the new trusted systems database. An error message will be displayed on `stderr` and the program will exit with an exit status of 1.

Note 2

This result is ambiguous because a “*” is returned both when the shadow password file cannot be opened and when the caller is not `UID 0`.

(But you can tell the difference from `errno`: it is set to `ENOENT` if the file can't be opened, and to `EACCES` if the caller is not `UID 0`.)

Note 3

These routines will return `NULL` because they can't open the pre-10.0 shadow password file `/.secure/etc/passwd`, which does not exist on 10.x. If the application is linked with the `ld(1)` `-z` option and the `NULL` return value is dereferenced, the program will get a `SIGSEGV` signal. Unless the program has a `SIGSEGV` signal handler, it will abort with a core dump.

Note 4

`fgetspwent(3C)` and `putspwent(3C)` will not work on 10.x if the calling application uses them for reading or updating `/.secure/etc/passwd`, because that file no longer exists. Refer to the 10.x `getprpnam(3)` manpage for replacement routines.

If the application uses these routines to maintain a password file *other than* `/.secure/etc/passwd`, these routines will work on 10.x as they did before.

See the section on Commercial Security in the *Release Notes* for more information.

8-10 Compatibility between 9.x Releases and 10.01

HP-UX (“DUX”) Clusters No Longer Supported

HP-UX Clusters are no longer supported as of 10.0.

The “DUX” and context-dependent file routines have been stubbed out in 10.x. The results of calling these routines are summarized in the table that follows. Note the special case of `getcontext()`, which is an entirely new routine as of 10.0.

Routine	Linked shared (uses 10.x libc)	Linked archived (used pre-10.0 libc)
=====		
cnodes	returns 0	returns 0
cnodeid	returns 1	returns 1
setccent	returns void; errno set to ENOENT	returns void; errno set to ENOENT
endccent	returns void; errno set to ENOENT	returns void; errno is unchanged
fgetcctent		
getcccid	return NULL;	return NULL;
getcctent	errno set to ENOENT	errno set to ENOENT
getccnam		
ftwh	returns -1; errno set to EOPNOTSUPP	works like ftw
getcdf	returns NULL; errno set to EOPNOTSUPP	returns NULL; errno set to ENOENT
getcontext	gets SVID3-compliant (different) behavior	returns -1; errno set to EOPNOTSUPP
getcwd	returns NULL; errno set to EOPNOTSUPP	works like getcwd
hidecdf	returns NULL; errno set to EOPNOTSUPP	context-dependent components are not hidden
mkrnod	returns -1; errno set to EOPNOTSUPP	returns -1; errno set to EOPNOTSUPP
nftwh	returns -1; errno set to EOPNOTSUPP	works like nftw

8-12 Compatibility between 9.x Releases and 10.01

Remedy: No direct replacements.

NLS Routines No Longer Supported

The following Native Language Support routines and macros are not present in the 10.x libc:

byte_status(3C)	nl_isalpha(3C)
c_colwidth(3C)	nl_iscntrl(3C)
catgetmsg(3C)	nl_isdigit(3C)
catread(3C)	nl_isgraph(3C)
currlangid(3C)	nl_islower(3C)
firstof2(3C)	nl_isprint(3C)
fprintmsg(3C)	nl_ispunct(3C)
getmsg(3C)	nl_isspace(3C)
ICONV(3C)	nl_isupper(3C)
iconvclose(3C)	nl_isxdigit(3C)
iconvlock(3C)	nl_msg(3C)
iconvopen(3C)	nl_printf(3C)
iconvsize(3C)	nl_scanf(3C)
idtolang(3C)	nl_sprintf(3C)
langinfo(3C)	nl_sscanf(3C)
langinit(3C)	nl_strcmp(3C)
langtoid(3C)	nl_strncmp(3C)
nl_asctime(3C)	nl_strtod(3C)
nl_ascxtime(3C)	nl_tolower(3C)
nl_atof(3C)	nl_toupper(3C)
nl_catopen(3C)	printmsg(3C)
nl_ctime(3C)	secof2(3C)
nl_cxtime(3C)	sprintmsg(3C)
nl_fprintf(3C)	strcmp16(3C)
nl_fscanf(3C)	strcmp8(3C)
nl_gcv(3C)	strncmp16(3C)
nl_init(3C)	strncmp8(3C)
nl_isalnum(3C)	

When linked into an archived 9.x executable and run on 10.x, these routines will behave the same as on 9.x (in conjunction with 9.x locales, which are still supported on 10.x; these routines will not support the 10.x locales).

When these routines are linked into a shared 9.x executable, the application will abort on 10.x, because the routines are not in `libc`. The abort message will say something like:

```
/usr/lib/dld.sl: Unresolved symbol: catgetmsg (code) from ./foo
Abort(coredump)
```

Remedy: See the section on NLS in the chapter on commands and libraries in the *Release Notes* for alternative routines.

8.x Applications Linked Archived. These will not work in internationalized mode, because the 8.x locales are not provided on 10.x. But you can copy the 8.x locales onto the 10.x system if you need them.

Other NLS Changes. See the section on NLS in the chapter on commands and libraries in the *Release Notes*.

NetIPC Routines No Longer Supported

The following Network Interprocess Communication (NetIPC) routines are not present in the 10.x `libc`:

<code>addopt(3N)</code>	<code>ipcname(3N)</code>
<code>initopt(3N)</code>	<code>ipcnamerase(3N)</code>
<code>ipconnect(3N)</code>	<code>ipcrecv(3N)</code>
<code>ipcontrol(3N)</code>	<code>ipcrevcn(3N)</code>
<code>ipcreate(3N)</code>	<code>ipcselect(3N)</code>
<code>ipcdest(3N)</code>	<code>ipcsend(3N)</code>
<code>ipcerrmsg(3N)</code>	<code>ipcsetnodename(3N)</code>
<code>ipcerrstring(3N)</code>	<code>ipcshutdown(3N)</code>
<code>ipcgetnodename(3N)</code>	<code>optoverhead(3N)</code>
<code>ipclookup(3N)</code>	<code>readopt(3N)</code>

The results of calling these routines are summarized in the table that follows.

Routine	When linked shared (uses 10.x libc)	When linked archived (used pre-10.0 libc)
=====		
adopt		
initopt		
ipcerrmsg	Abort (note 1)	9.x behavior
ipcerrstring		
optoverhead		
readopt		
ipconnect		
ipcontrol		
ipcreate		
ipcdest		
ipcgetnodename		
ipclookup		
ipcname	Abort (note 1)	SIGSYS signal/abort (note 2)
ipcnamerase		
ipcrecv		
ipcrecvcn		
ipcselect		
ipcsend		
ipcsetnodename		
ipcshutdown		

NOTES:**Note 1:**

The application will be abort because the routines are not in `libc`. The abort message will say something like:

```
/usr/lib/dld.sl: Unresolved symbol: addopt (code) from ./foo
Abort(coredump)
```

Note 2:

The application will receive a SIGSYS signal when one of these routines is called. Unless the program has SIGSYS signal handler, it will terminate with a core dump. The abort message will say:

```
Bad system call(coredump)
```

Remedy: Use BSD Sockets instead of NetIPC. See NetIPC to BSD Sockets Migration Guide (HP part number 98194-90045). For additional ARPA transport information, see the manual Installing and Administering LAN/9000 Software (HP part number 98194-90050).

Changes in Error Handling for Math Library Functions

As of Release 10.0, error handling for math library functions has been updated where necessary to conform to the latest version of the X/Open Portability Guide (XPG4). The math library manpages describe the 10.x error handling for each function in detail.

Summary of Changes.

- There is no longer any difference between error handling in `libm` (the SVID library) and `libM` (the XPG4 library).

XPG4 is a superset of SVID3, and the two standards do not conflict. The most visible change is that the `libm` library functions no longer produce `DOMAIN`, `SING`, `PLOSS`, or `TLOSS` error messages.

- If XPG4 does not require a function to set `errno`, the function does not set it.

For example, most library functions used to set `errno` to `EDOM` if a function argument was a `NaN` (Not-a-Number); they no longer do so, though they continue to return a `NaN`.

The table that follows describes the changes in error handling for each function.

Error handling is described in terms of the return value and the `errno` setting. For example, `NaN`, `EDOM` means that `NaN` is returned and `errno` is set to `EDOM`.

Function	9.0 libm	9.0 libM	10.0
=====			
Most, NaN args	NaN, EDOM	NaN, EDOM	NaN, 0
Most, underflow	0, ERANGE	0, ERANGE	0,0
(scalb is the only function still required to set errno to ERANGE in the case of underflows.)			
acos(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
acos(x >1.0)	0, EDOM	NaN, EDOM	0, EDOM
acosd(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
acosd(x >1.0)	0, EDOM	NaN, EDOM	0, EDOM
acosf(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
acosf(x >1.0)	NaN, EDOM	NaN, EDOM	0, EDOM
acosdf(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
acosdf(x >1.0)	NaN, EDOM	NaN, EDOM	0, EDOM
acosh: no change except for NaN arguments			
asin(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
asin(x >1.0)	0, EDOM	NaN, EDOM	0, EDOM
asind(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
asind(x >1.0)	0, EDOM	NaN, EDOM	0, EDOM
asinf(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
asinf(x >1.0)	NaN, EDOM	NaN, EDOM	0, EDOM
asindf(+/-INF)	NaN, EDOM	NaN, EDOM	0, EDOM
asindf(x >1.0)	NaN, EDOM	NaN, EDOM	0, EDOM
asinh: no change except for NaN arguments			
atan: no change except for NaN arguments			

8-18 Compatibility between 9.x Releases and 10.01

Function	9.0 libm	9.0 libM	10.x
=====			
atan2(0,0)	0, EDOM	0, EDOM	NaN, 0
atan2d(0,0)	0, EDOM	0, EDOM	NaN, 0
atan2df(0,0)	0, EDOM	0, EDOM	NaN, 0
atan2f(0,0)	0, EDOM	0, EDOM	NaN, 0
atand: no change except for NaN arguments			
atandf: no change except for NaN arguments			
atanf: no change except for NaN arguments			
atanh(+/-1)	NaN, EDOM	NaN, EDOM	NaN, 0
cabs(overflow)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	HUGE_VAL, 0
cbirt(negative)	0, EDOM	0, EDOM	result, 0
cbirtf(negative)	0, EDOM	0, EDOM	result, 0
copysign: no change			
copysignf: no change			
cos(large arg)	0, ERANGE	0, ERANGE	result, 0
cos(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
cosd(large arg)	0, ERANGE	0, ERANGE	result, 0
cosd(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
cosdf(large arg)	0, ERANGE	0, ERANGE	result, 0
cosdf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
cosf(large arg)	0, ERANGE	0, ERANGE	result, 0
cosf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0

Function	9.0 libm	9.0 libM	10.x
=====			
cosh: no change except for NaN arguments			
coshf: no change except for NaN arguments			
drem(x, 0)	NaN, 0	NaN, 0	NaN, EDOM
drem(+/-INF, y)	NaN, 0	NaN, 0	NaN, EDOM
exp(+INF)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	+INF, 0
exp(-INF)	0, ERANGE	0, ERANGE	0, 0
expf(+INF)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	+INF, 0
expf(-INF)	junk, 0	+INF, 0	0, 0
fabs: no change except for NaN arguments			
fabsf: no change except for NaN arguments			
finite: no change			
finitef: no change			
floor: no change except for NaN arguments			
fmod(x, 0)	x, 0	NaN, EDOM	NaN, EDOM
fmodf(x, 0)	x, 0	NaN, EDOM	NaN, EDOM
hypot(overflow)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	HUGE_VAL, 0
isinf: no change			
isinff: no change			
isnan: no change			
isnanf: no change			
j0(large values)	0, ERANGE	0, ERANGE	result, 0
j1(large values)	0, ERANGE	0, ERANGE	result, 0
jn(large values)	0, ERANGE	0, ERANGE	result, 0

8-20 Compatibility between 9.x Releases and 10.01

Function	9.0 libm	9.0 libM	10.x
lgamma(int <= 0)	HUGE_VAL, EDOM	HUGE_VAL, EDOM	HUGE_VAL, 0
lgamma(overflow)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	HUGE_VAL, 0
lgamma(+INF)	NaN, ERANGE	NaN, EDOM	NaN, 0
lgamma(-INF)	HUGE_VAL, ERANGE	HUGE_VAL, ERANGE	HUGE_VAL, 0
log(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
log(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	NaN, EDOM
log10(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
log10(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	NaN, EDOM
log10f(0)	-MAXFLOAT, EDOM	-MAXFLOAT, EDOM	-MAXFLOAT, 0
log10f(negative)	-MAXFLOAT, EDOM	+INF, EDOM	NaN, EDOM
log2(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
log2(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	NaN, EDOM
log2f(0)	-MAXFLOAT, EDOM	-MAXFLOAT, EDOM	-MAXFLOAT, 0
log2f(negative)	-MAXFLOAT, EDOM	-MAXFLOAT, EDOM	NaN, EDOM

Function	9.0 libm	9.0 libM	10.x
=====			
logf(0)	-MAXFLOAT, EDOM	-MAXFLOAT, EDOM	-MAXFLOAT, 0
logf(negative)	-MAXFLOAT, EDOM	+INF, EDOM	NaN, EDOM
logb(0)	-INF, 0	-INF, 0	-HUGE_VAL, EDOM
modf: no change			
pow(-INF, +even int)	1, 0	1, 0	+INF, 0
pow(-INF, +odd int)	-1, 0	-1, 0	-INF, 0
pow(-INF, -even int)	1, 0	1, 0	0, 0
pow(-INF, -odd int)	-1, 0	-1, 0	0, 0
pow(0, 0)	0, EDOM	1, 0	1, 0
pow(0, negative)	0, EDOM	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM
pow(negative, +-INF)	0, EDOM	NaN, EDOM	NaN, 0
pow(neg., non-int)	0, EDOM	NaN, EDOM	NaN, EDOM
powf(0, 0)	NaN, EDOM	NaN, EDOM	1, 0
powf(0, negative)	NaN, EDOM	NaN, EDOM	-HUGE_VAL, EDOM
powf(x>1, +INF)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	+INF, 0
powf(x>1, -INF)	0, EDOM	0, EDOM	+INF, 0
powf(0<x<1, -INF)	0, EDOM	0, EDOM	0, 0
powf(0<x<1, +INF)	HUGE_VAL, EDOM	HUGE_VAL, EDOM	0, 0
powf(0, negative)	NaN, EDOM	NaN, EDOM	-HUGE_VAL, EDOM
powf(negative, +INF)	NaN, EDOM	NaN, EDOM	NaN, 0
powf(negative, -INF)	-INF, 0	-INF, 0	NaN, 0
rint: no change except for NaN arguments			

8-22 Compatibility between 9.x Releases and 10.01

Function	9.0 libm	9.0 libM	10.x
scalb(overflow)	+INF, 0	+INF, 0	+HUGE_VAL, ERANGE
scalb(underflow)	0, 0	0, 0	0, ERANGE
sin(large arg)	0, ERANGE	0, ERANGE	result, 0
sin(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
sind(large arg)	0, ERANGE	0, ERANGE	result, 0
sind(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
sindf(large arg)	0, ERANGE	0, ERANGE	result, 0
sindf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
sinf(large arg)	0, ERANGE	0, ERANGE	result, 0
sinf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
sinh: no change except for NaN arguments			
sinhf: no change except for NaN arguments			
sqrt(negative)	0, EDOM	NaN, EDOM	NaN, EDOM
sqrtf: no change			
tan(large arg)	0, ERANGE	0, ERANGE	result, 0
tan(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
tand(large arg)	0, ERANGE	0, ERANGE	result, 0
tand(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
tandf(large arg)	0, ERANGE	0, ERANGE	result, 0
tandf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0

Function	9.0 libm	9.0 libM	10.x
=====			
tanf(large arg)	0, ERANGE	0, ERANGE	result, 0
tanf(+/-INF)	0, ERANGE	NaN, EDOM	NaN, 0
tanh: no change except for NaN arguments			
tanhf: no change except for NaN arguments			
y0(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	-HUGE_VAL, 0
y1(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	-HUGE_VAL, 0
yn(negative)	-HUGE_VAL, EDOM	-HUGE_VAL, EDOM	-HUGE_VAL, 0
y0(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
y1(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
yn(0)	-HUGE_VAL, EDOM	-HUGE_VAL, ERANGE	-HUGE_VAL, 0
y0(large values)	0, ERANGE	0, ERANGE	result, 0
y1(large values)	0, ERANGE	0, ERANGE	result, 0
yn(large values)	0, ERANGE	0, ERANGE	result, 0

If a program depends on the 9.x math library error handling, it may behave differently on a 10.x system if it uses shared libraries or if it is recompiled on 10.x. A program that linked in an archive library on a pre-10.0 system will continue to behave as it did before.

8-24 Compatibility between 9.x Releases and 10.01

Floating-Point to Integer Conversion

The behavior of an overflow exception on a floating-point convert-to-integer operation has been modified so as to be consistent on the Series 700 and 800, and to comply with the IEEE Standard and the behavior documented in the *HP-UX Floating-Point Guide*, HP part number B3906-90003.

In previous releases, the Series 800 took a floating-point exception on this conversion even if all traps were disabled. In 9.x, the Series 700 did not generate a floating point exception; instead the result was truncated to `MAXINT/MININT` and the `OVERFLOW` bit was set.

Both of these behaviors were incorrect according to the IEEE specification, and did not accord with what was documented in the *HP-UX Floating-Point Guide*.

The correct behavior is not to trap with a floating-point exception if traps are disabled, but return `MAXINT/MININT`, set the `INVALID` bit, and raise an `FP` exception only if the `INVALID` trap-enable bit is on. This is the behavior of HP-UX 10.x on both the Series 700 and 800.

You can enable traps for some or all of these conditions by using the `fpsetdefaults` function, the `fpsetmask` function, the `+T` option (`£77` only), or the `+FP` compiler option. For more information, see Chapter 5 of the *HP-UX Floating-Point Guide*.

Other Changes Affecting Applications

Standards Conformance Changes. The following routines have been changed in 10.x for standards conformance.

Routine	Reason
=====	
iconv(3C)	XPG4.2 conformance: function now returns <code>size_t</code> rather than <code>ssize_t</code> .
nftw(3C)	XPG4.2 conformance: passes <code>FTW_SLN</code> rather than <code>FTW_SL</code> for a dangling symlink. see the entry for this command in Chapter 6 of the <i>Release Notes</i> for more information.
openlog(3C) syslog(3C) closelog(3C)	XPG4.2 conformance: functions now return void rather than int. Note: function prototypes for these functions were not delivered before 10.0.
tcsendbreak(3C)	SVID3 conformance: the "duration" parameter is no longer ignored -- its value is used to alter the function's behavior.

I/O Convergence Changes. The following routine has been changed to support I/O convergence (see the section on IO convergence in the *Release Notes*):

dial(3C)	Device name suffix was 3 characters on 9.x Series 700; 4 characters on 9.x Series 800. It is 4 characters on 10.x.
----------	--

Linker Defects

The `/bin/ld` and `/lib/dld.sl` released at 9.0 had a pair of defects that could cause a program that runs correctly on a 9.x system to work incorrectly on 10.x systems. Fixes for these defects are included in the "Upgrade Tools"

8-26 Compatibility between 9.x Releases and 10.01

package that includes this manual; see “Locating and Loading Tools and Documentation” in Chapter 4.

The problem could result in the wrong version of a symbol being used; or, if the version which would be used is unavailable, an unsatisfied symbol error may occur.

The defect occurs when a shared library is built with an explicit dependency on a second shared library that uses versioned symbols. The summary version information recorded in the first shared library can be incorrect. This can cause problems for any program or shared library which uses the first shared library.

Affected programs or shared libraries should be re-linked on a 9.x system using the latest linker patch, (PHSS_5290 or later) using the shared library versions that were current when the program or the shared library with the problem was built.

Note

- Use the tools described below to identify and resolve problems *before* upgrading a system to 10.01.

When libraries are updated during the upgrade, information may be lost which can identify with certainty if a program or shared library has a problem.

- You *do not* need to worry about linker problems if:
 - Your applications do not use shared libraries.
 - You know for certain that that your shared libraries do not use versioning or dependent libraries.

If in doubt, run the `chksysver` program described below.

You should re-link affected programs or shared libraries *on a 9.x system* using the tools provided in this package (see “Locating and Loading Tools and Documentation” in Chapter 4), and the shared library versions that were current when the program or the shared library with the problem was built.

Don’t re-link with newer versions of the shared libraries; if you do, the re-link will cause the program or shared libraries to pick up the newer versioned symbols, which may not be what the program or shared library needs.

The tools are:

- The program `chkversion`.

This indicates whether the programs or shared libraries you specify are affected by this problem.

- The script `chksysver`.

This examines all of the programs and shared libraries on a system.

See the `chkversion(1M)` and `chksysver(1M)` manpages (included in this package) for more details.

Procedure. Do tasks in the following order:

1. Load the tools from the upgrade tape or disk as described in Chapter 4.
2. Run `chksysver(1M)` to check all the shared libraries and programs on the system.

This could take from one to two hours. (Example times: 45 minutes on a 755 using 4Gb of data. 1 hour and 45 minutes on a 720 using 3Gb of data.)

If you know that only specific programs or libraries could have the problem, use `chkversion(1M)` instead. See the manpages for more information.

3. If `chksysver` or `chkversion` reports any problems:

- Load the new version of `dld.sl` from the PHSS_5290 patch.
- Relink the affected programs and libraries.

You must do this as soon as you have downloaded the new version of `/lib/dld.sl`; otherwise the affected code will now encounter the same errors when you try to run it on 9.x as it would on 10.01.

Re-link shared libraries with the patched linker, then re-link any shared libraries or programs which depend upon the corrected shared library. If you re-link a shared library, re-link the programs and shared libraries which depend upon it.

For example, if the report generated by `chksysver` says,

```
Y has incorrect version information
```

—then you need to re-link Y.

If the report says,

```
X has incorrect version information for Y
```

—then you need to re-link X (you should have already re-linked Y).

- Log in as root and run a script to correct version-propagation behavior in the affected libraries:

```
/usr/contrib/bin/install_verfix -uninstall
```

See the `chksysver(1M)` manpage for more information.

Technical Details. The highwater mark of a shared library should be the maximum of the highest version number of any symbol defined by the shared library and the highwater mark of any of the shared libraries mentioned on the link line which created the shared library.

A defect in the 9.0 `/bin/ld` caused the highwater marks of the dependent libraries to be ignored, which can cause an incorrect highwater mark to be recorded in a shared library. This can cause problems for a program or another shared library which uses a shared library with the wrong highwater mark.

A defect in `/lib/dld.sl` allowed these defective programs to run without an apparent error.

Moving Files

Because the filesystem layout was changed when you upgraded to 10.01, (see “The HP-UX 10.0 File System” in the *Release Notes for HP-UX 10.10* for more information) you need to be particularly careful about moving files from a 9.x system to a 10.x system.

“Moving files” means recovering backups, transferring files over a network via NFS or `ftp`, or mounting 9.x filesystems from a remote system or from a disk that you have moved from one system to another.

Recovering, Moving or Mounting Files from a 9.x System

The following are guidelines for recovering files from 9.x backups onto a 10.x system. They also apply to transferring files over a network via NFS or `ftp`, or mounting 9.x filesystems from a remote system or from a disk that you have moved from one system to another.

- Unless you’re sure of what’s on the tape (or disk, etc.), don’t do a blanket recover (mount, etc.) onto a 10.x system.

Make sure you recover or mount only “user”, as opposed to “structural”, files and directories. (“Structural” files and directories are those that form part of HP-UX, such as `/` (root), `/bin`, `/usr/bin` and `/usr/lib` on 9.x systems.)

If you really need to restore a file from one of these “system” directories from a 9.x tape onto a 10.x system (for example, to retrieve non-HP custom code in a script) one way to do it could be to recover the file to a different pathname; for example, to recover the 9.05 version of `/etc/inittab` from a tape created with `fbbackup(1M)`, you might create the directory `/9.05` on your 10.x system and recover the file to `/9.05/etc/inittab`.

Caution

Be very careful if you attempt this.

The above example will work if you are using `frecover(1M)` and you use the `-X` option to recover the file to the current working directory (which should be `/9.05` in this example).

But if you used `tar(1)` or `cpio(1)` to back up files to absolute pathnames (that is if the `inittab` file was backed up as `/etc/inittab` rather than `etc/inittab` or `./etc/inittab`), then you *cannot* recover them to a different pathname; in that case the `/etc/inittab` file on the 9.x tape would overwrite the 10.x version of the file, which is just what you don't want.

If you are not sure if files on a tape were backed up to absolute pathnames or not, you can make the recovery utility print a table of tape contents; for example, use the `t` option to `tar` or `cpio` (see the `tar(1)` and `cpio(1)` manpages for more information).

- Avoid recovering HP-created context-dependent files (CDFs) onto 10.x systems; these are “structural” files and will not work on 10.x (see the guidelines for “structural” files above).

If you do recover a CDF onto 10.x, you'll find that the “hidden directories” that comprise CDFs are treated as ordinary directories. For example, the CDF element `/myapp/outputfile+/node1` will become an ordinary file (`node1`) in the ordinary directory `/myapp1/outputfile+` when you use `tar` or `frecover` to restore it to a 10.x system.

If you mount a disk containing CDFs onto a 10.x system, you won't see the plus sign in the directory name, but the sticky bit will be set.

- Be careful about recovering *any* files or directories from 9.x backups, or moving or mounting them from a 9.x system.

If you follow filesystem-layout guidelines in the *HP-UX System Administration Tasks* manual and the *HP-UX 10.0 File System Layout White Paper*, even your “user” files are likely to be in different directories on the 10.x system from where they were on 9.x. Recovering these files blindly may create a redundant set of directories; the next run of your application may not find the data you recovered and produce erroneous results.

If in doubt, always recover 9.x files to a temporary directory, not to absolute pathnames, if the files have been backed up in a way that makes this possible (see the discussion above). Then either copy the files you need to their correct 10.x locations, or (if these are system files) extract the information you need and merge it into the corresponding 10.x files.

A

Contents of the HP-UX 10.1 “Core Media”

Contents of 10.01

The following table lists the bundles and products contained on the HP-UX 10.01 Core OS “media” (tape or CD).

The items in boldface are HP-UX **bundles**. The Desktop and Runtime bundles (**HPUXEngRTx00**, etc., marked with an asterisk) are supplied in the various languages (French, German, Spanish, Italian, Japanese, etc.) but are not all listed here because of space limitations.

The user license bundles are no longer part of the core media. Many of the products contained on the core media are also included in application media. This list does not contain products or bundles that require additional purchase.

Table A-1. HP-UX 10.0 Bundles/Products

Bundle/Product Name	Size (KBytes)	Description
2UserDegrade	95	HP-UX 2-User License (for degrading user license level)
Accounting	446	Accounting
Analysis-Tools	4560	HP-UX 9.0-10.0 Upgrade Analysis Tools Bundle
AudioSubsystem	4799	HP-UX Audio Subsystem
Curses-Color	1498	Curses library with color support
DCE-Core	7929	HP DCE/9000 Core Client Software
DCE-CoreTools	9978	HP DCE/9000 Application Development Tools
DCEProg	9981	DCE Programming and Archive Libraries Bundle
DFS-Core	3435	HP DCE/9000 Distributed File System
DesktopConfig	2	Sys. config.-Desktop HP-UX single-user sys. (Ser. 700 only)
DiskQuota	170	DiskQuota
Diskless	25572	Diskless (needed to set up NFS Diskless Cluster)

A-2 Contents of the HP-UX 10.1 “Core Media”

Table A-1. HP-UX 10.0 Bundles/Products (continued)

Bundle/Product Name	Size (KBytes)	Description
GraphicsCommon	13768	HP-UX Common Graphics Libraries (Ser. 700 only)
GraphicsPEX5v2RT	3884	HP-UX PEX 5.1 Version 2 (Ser. 700 only)
HPUXEngDT700*	235399	English HP-UX Desktop Environment (Ser. 700)
HPUXEngGS800*	289323	English Run-time HP-UX Environment (Ser. 800)
HPUXEngRT700*	303233	English HP-UX Run-time Environment (Ser. 700)
HPUXEngRT800*	181415	English HP-UX non-Graphics Runtime Environment (Ser. 800)
HPUX-Install	21628	HP-UX System Installation Services
ImagingSubsystem	2806	HP-UX Image Viewer Subsystem
InstantIgnite	664	Instant Ignition on Series 700
InstantIgnite	15	Instant Ignition on Series 800
Integ-Logon	6101	Integrated Logon Bundle
IntegratedLogin	6029	Integrated Login for HP-UX and DCE
International	10574	International
InternetSrvcs	4304	General network applications and daemons
JournalFS	2290	The Base VxFS File System
Keyshell	600	Keyshell
LSSERV	1967	Network Licensing Server Software
LVM	3674	LVM
MSDOS-Utills	148	MSDOS-Utills
MailUtilities	1008	User mail agents and related tools
NCSNCK	2081	NCS Network Computing Kernel
NFS	2880	ONC/NFS; Net. File Sys.,Info. Services,Utilities
NetInstall	24202	Network Cold Install Utilities
Networking	6324	HP-UX_10.0_Lanlink_Product
NonHP-Terminfo	499	Non HP terminfo files
OS-Core	64825	Core Operating System
PEXDemos	3863	PEX Demos (Ser. 700 only)
PrinterMgmt	1033	PrinterMgmt

Table A-1. HP-UX 10.0 Bundles/Products (continued)

Bundle/Product Name	Size (KBytes)	Description
Proc-Resrc-Mgr	157	HP Process Resource Manager
ProgSupport	15561	ProgSupport
SCSI-Passthru	84	SCI-Passthru (Ser. 800 only)
SDF-Tool	392	HP SDF Tools (Ser. 800 only)
SDF-Tools	399	SDF-Tools (Ser. 800 only)
SOE	659	SOE
SUPPORT-TOOLS	19	HP SUPPORT TOOLS
SW-DIST	5135	HP-UX Software Distributor
ScreenCapture	1181	HP-UX Screen Capture
SecurityMon	714	SecurityMon
SharedPrint	8678	HP-UX Shared Print Services
SourceControl	1157	SourceControl
Spelling	243	Spelling
Streams	1710	HP-UX_10.0_Streams_Product
Streams-TIO	206	HP-UX_10.0_Streams-TIO_Product
SystemAdmin	14124	HP-UX System Administration Tools
SystemComm	1394	Sys. Comm. utilities - ct,cu,ptydaemon,vt,kermit
TerminalMngr	1977	TerminalMngr
TextEditors	444	TextEditors
TextFormatters	1177	TextFormatters
UUCP	1028	Unix to Unix CoPy
Upgrade	4612	Upgrade
UserLicense	34	HP-UX User License
VT3K	83	VT3K -Provides virtual terminal from s700/800 to HP3000
X11	93198	HP-UX X Window Software
X11BMS	889	HP-UX BMS for X Window Software

A-4 Contents of the HP-UX 10.1 “Core Media”

B

NFS Diskless Questions and Answers

This appendix answers some common questions about administering NFS Diskless. It is a slightly condensed version of the “Questions and Answers” section of the *NFS Diskless Concepts and Administration*, which is supplied in its entirety in postscript format on the “Upgrade Tools” tape or CD packaged with this manual (see “Locating and Loading Tools and Documentation” in Chapter 4).

Cluster Configuration

Question: I have 100 clients and they all have the same kernel. Can the kernels be linked to save disk space?

Answer: Yes. Client kernels (`/export/tftpboot/client/stand/vmunix`) can be hard linked with each other to save disk space.

It is also possible to hard link both the RAM file system (`vmunix.fs`) and the LIF volume (`uxbootlf`). By default, the `vmunix`, `vmunix.fs`, and `uxbootlf` files are hard linked with identical files used by other clients. After the initial boot, the `vmunix` file is rebuilt by the client and the link is broken.

Use only hard links, because:

- Symbolic links do not work for kernels.

This is because the `/export/tftpboot/client/stand` directory is mounted as `/stand` on a client system and any symbolic links within this directory are resolved in the context of the client, not the server.

- HP does not support symbolic links for linking boot files.

Operations which modify the kernel or other boot files break any existing links before writing a new boot file. This prevents a change to one client's boot file from affecting all clients that may have been linked with that boot file. The best way to change all the clients' boot files is to change a single client and then re-establish the hard link.

Question: I have 100 clients, and they won't all fit on the same disk.

Answer: You can spread the clients' private directories and boot file directories across multiple volumes. You can do this in one of two ways:

1. Before the client is added.

When you add a client via SAM, SAM creates two directories to hold the client's private files:

- the private root `/export/private_roots/client`

- the boot file or kernel directory
`/export/tftpboot/client`.

You can create these directories “by hand”(not using SAM), before adding a client. The directories must be empty when you use SAM to add the client.

So long as it finds these directories empty, the SAM cluster configuration code will honor them and put the appropriate files in them. If the directories are not empty, the cluster configuration code exits with an error.

Creating the client directories ahead of time allows you to redirect them to a volume with more disk space. You can do this by means of:

- A symbolic link to an empty directory on another volume.
- A mounted physical volume (the existence of a `lost+found` directory does not affect the empty status).
- A mounted logical volume.

2. After the client is added.

After a client has been added, you can still move the client directories to another volume and use symbolic links to link the old directory to the new.

Note When you use SAM to remove a client and its files, if either the private root or kernel directory is a symbolic link, SAM will remove the target of the link, but not the link itself.

Question: I want `/usr/bin` to be a separate file system on my server.

Answer: This presents a problem.

Sharing between server and client is done by a mechanism called **share links**. Under HP-UX, share links are defined at `/usr`, `/sbin`, and several directories under `/opt`.

As a result, NFS mount points are established for `/usr`, `/sbin`, and the `/opt` directories on the client. If a subdirectory of a sharing point (a directory specified as a share link) is

a separate file system, the file-sharing model breaks down because NFS does not propagate the mount point.

If you set up your server so that a subdirectory of a sharing point is a separate file system, you must export this file system (from the system that serves it) and mount it (on the client) “by hand” (or you can use SAM by to make the subdirectory a “cluster-wide” file system).

HP does not recommend this configuration; it may cause problems when you update from one release to the next, and possibly during other operations.

Question: I want `/usr/local` to be a separate file system on my server.

Answer: This is less of a problem because:

1. HP does not deliver anything to `/usr/local`
2. `/usr/local` is not necessary for booting a system.

If `/usr/local` is a separate file system on a server, you can use SAM to export it and make it a “cluster-wide” file system, mounted at `/usr/local` on all the clients.

Question: I added a remote client and now I cannot manage it via SAM.

Answer: When SAM adds remote clients, the only name service propagated to them is the `/etc/hosts` file. This is because any NIS and DNS configuration found on the server is likely to be inappropriate for a remote client on a different network. After adding a remote client, you must set up the client’s DNS or NIS configuration “by hand”.

Question: Can I spread client swap space among many disks?

Answer: Yes. By default, clients swap to `/paging` in their root file system. In other words, a given client swaps to `/export/private_roots/client/paging` on the server. But clients can also swap to other remote file systems via NFS, or to a local disk via device swap or filesystem swap.

B-4 NFS Diskless Questions and Answers

Note You cannot configure swap for NFS Diskless clients into their kernels; you must do it either by running `swapon` from the command-line, or through entries in the client's `/etc/fstab`.

Apart from this limitation, a client has the same choices for swap as the server.

If you want the client to swap to some other destination than `/paging`, remove the `swapfs` entry for `/` in the client's `/etc/fstab`.

Caution When default swap to `/paging` is disabled, the client does not have any swap space as it begins to boot; it swaps to RAM until a swap device can be configured from entries in its `/etc/fstab`. For this to work, the client must have enough available RAM to boot to the point when the `swapon` command is executed.

Swap entries for local disks or file systems are processed early in the boot process, but NFS-based swap entries are processed at the end of the boot process.

This means that primary swap for a client *must* be either the default swap to `/paging`, or swap to a local device or file system. HP does not support primary swap to a remote file system other than `/paging`. Use other remote file systems only for auxiliary swap.

Question: I notice that the `/.rhosts` files on my server and clients allow for root equivalence throughout the cluster. Can I remove these entries?

Answer: No. Single-point administration via SAM depends on root equivalency throughout the cluster. If you remove client or server entries from the `/.rhosts` files you will not be able to use SAM to administer the cluster.

Question: How do I “clusterize” my system?

Answer: An NFS server does not need to be “clusterized”.

HP's proprietary "DUX" technology, which NFS Diskless replaces, required a configuration process on the server which converted key system files (`hp-ux`, `/etc/checklist`, and others) into context dependent files (CDFs) and modified the server's kernel to enable diskless functions.

NFS Diskless does require any modification of the server's file system. The only side effects of adding a diskless client are:

- Various boot services (BOOTP, TFTP, rbootd, NFS, etc) are configured.
- The kernel is configured for NFS (if necessary).

These tasks are performed automatically when you use SAM to add the first client to your system.

Question: Do I install software for clients onto the client or the server?

Answer: Software for diskless clients should be installed on the server, and propagated to clients using NFS mounts.

Use the `swcluster(1M)` command to do this. `swcluster` runs other Software Distributor (SD) commands such as `swinstall` to set up NFS mounts, and creates `/etc/fstab` entries to NFS-mount the proper directories from the diskless server to the clients. It also creates an "installed product database" (IPD) on the client which tracks installed software and allows it to be configured.

Question: Once I have a cluster, how do I install additional software?

Answer: Use `swcluster(1M)`. See "Software Administration" in section 4 of the *NFS Diskless Concepts and Administration* white paper for more information.

Question: Who creates shared roots and how are they named?

Answer: When the SAM fileset (`SystemAdmin.SAM`) is installed in the server's root directory, it creates a shared root named `/export/shared_roots/OS_700` (if this is a Series 700 system). This shared root is a symbolic link to the root directory (`/`).

As the system administrator, you can create other shared roots with any name you choose, although HP recommends certain

B-6 NFS Diskless Questions and Answers

name elements: *architecture, application vs. OS, release level*. You can create these directories “by hand”, and they are also created by `swinstall` when you perform an **alternate root install**.

An alternate root install populates a shared root with sharing points (i.e. products). See `swinstall(1M)` for details.

Question: My client won't boot. What could be wrong?

Answer: If the client is not booting because the server simply does not respond to the client, the problem may be:

- The client's `bootptab` record may be incorrect or missing.

Check the `/etc/bootptab` file to make sure that the client's hardware address is specified correctly.

- The `bootp` daemon is incorrectly configured.

See “Boot Service Setup” in section 4 of the *NFS Diskless Concepts and Administration* white paper.

- `rbootd` daemon not running.

If the client has RMP-protocol boot ROMs, check to see if `rbootd` is running on the server. If it is not, start it and confirm that the `RBOOTD_START` parameter in `/etc/rc.config.d` is set to 1.

Note You may need to restart `rbootd` after configuring a new LAN card.

- The `/export/tftpboot/client/stand/uxbootlf` file is missing.

This is the file that is specified in the “bf” field in the client's `bootptab` record. If it is missing, `bootpd` will not respond to the client.

Fix: copy `/usr/lib/uxbootlf` to `/export/tftpboot/client/stand/uxbootlf`.

If the client is able to transfer its boot files, but fails at a later point, the problem may be with file system exports from the server. Check the client's `/etc/fstab` (`/export/private_roots/client/etc/fstab` on the server) against the file system exports on the server.

Use the `exportfs` command to see what is currently exported, or look at the `/etc/exports` file directly. If there is an error in the file, nothing may be exported.

Question: How can I tell what kind of boot ROM my system has?

Answer: You shouldn't need to know because both `bootp` and `rbootd` services are started on cluster servers. But, in general, any SPU model introduced in 1994 or later (starting with the 9000/712) has new (`BOOTP`) boot ROMs.

Performance

Question: How can I improve performance in my cluster?

Answer: There are several things that you can do to get better performance. Here are a few:

1. Use a local swap disk on each client.
2. Change the private root export to an asynchronous export.

If your clients swap to `/paging` in the private root, changing the private root export to an asynchronous export will considerably improve paging performance, but at a cost.

If the server crashes before a page from the client is committed to disk, one of the following may occur when the server comes back up:

- The process could crash when it pages back in.
- The process could cause silent data corruption.

You can change to an asynchronous export by editing the `/etc/exports` file on the server and

adding the `async` option to each private root export (`/export/private_roots/client`).

The following is an example of an entry in `/etc/exports` for an asynchronous private root export for a client named **zorro**:

```
/export/private_roots/zorro -async,root=zorro,access=zorro
```

3. Use multiple LAN cards on your server to spread the cluster over multiple networks.
4. Use FDDI on your server with an ethernet switch serving the 802.3 networks that the clients are connected to.

Question: My network becomes congested when booting many clients simultaneously. What can I do?

Answer: When many diskless clients boot from one boot server simultaneously, the server may be too busy to respond to each client's boot request quickly.

The default timeout values specified in each client's `/etc/fstab` file take into account large numbers of clients booting simultaneously. But your network traffic may vary and you may want to do one or both of the following:

- Adjust the timeout and retry values of the primary boot path for your clients. HP recommends setting the primary boot path so that your client does a directed boot request with infinite timeouts and retries.

Do this during the boot process on the client. If you have the older type of ROM (RMP protocol), enter administrator mode (press the escape key to interrupt the boot) and set the boot path as follows:

```
path pri lan.nnnnnn-nnnnnn.255.255
```

where `nnnnnn-nnnnnn` is the hardware address of the server system. If you have BOOTP protocol ROMS (newer systems), set your path as follows:

```
path pri lan.nnn.nnn.nnn.nnn.255.255
```

where *nnn.nnn.nnn.nnn* is the IP address of the server system.

- Distribute clients to multiple LANs to increase effective network bandwidth. If this still does not help, then a faster server (or more server RAM) may help.

Question: I built a new kernel on my diskless client and moved the old one to `vmunix.bak`. The new kernel doesn't boot for some reason, so I tried to boot the old one by interacting with ISL. My system panicked. What is wrong?

Answer: Two files are needed to boot an NFS diskless client: `vmunix` and `vmunix.fs`. If you attempt to boot a kernel named `XYZ`, the secondary loader will look for a second file named `XYZ.fs`. In this case, you need to make sure that `vmunix.bak` is accompanied by `vmunix.bak.fs`, or the system will not boot. A hard link or symbolic link to `vmunix.fs` should suffice.

If the corresponding `.fs` file is not present, the secondary loader will default to loading the file called "vmunix.fs". If this file is not compatible with your kernel, unexpected behavior (possibly bad) may result.

Single Point Administration

Policies

Question: I selected the **Shared Home Directories** policy, but my users' directories under `/users` and `/users2` did not appear on the clients. Is there a way to make them appear on all clients?

Answer: Yes, you can use SAM to make a file system cluster-wide.

Do this by selecting the file system and choosing the action **Manage as a Cluster-wide Resource** in SAM's **File Systems** area.

There are some things to consider:

B-10 NFS Diskless Questions and Answers

- If the file system is local to the server, then selecting **Manage Cluster-wide ...** will add the file system to the cluster clients using the automounter.

If you prefer not to use the automounter, execute SAM from a client and choose

Add a Remote File System -> Using NFS...

Set **Manage Cluster-Wide ...** to **Yes**.

- If the file system is not local to the server, add it to the server using NFS or the Automounter and set **Manage Cluster-Wide ...** to **Yes**.
- If the home directory is not a file system, but just a directory on the server, you need to perform the task *from a client*.

Add it to the client using NFS or the Automounter and set **Manage as a Cluster-Wide** to **Yes**.

Note

Use NFS (not Automounter) to mount a remote file system that will have a local file system mounted onto it. You should also include the mount option **boot** in the file system's entry in **/etc/fstab**. This ensures that the mounts occur in the correct order. (Local mounts occur after **boot** NFS mounts but before other NFS and automounter mounts).

Question: How do I configure local swap on clients?

Answer: Run SAM on each client that will have local swap and use the **Disk Devices** sub-area under **Disks and Filesystems** to add a local disk for swap.

- If you choose to add this device swap as primary swap, SAM will build a new kernel and reboot the client.

You may want to remove the NFS swap (**swapfs**) entry from **/etc/fstab**.

- If you choose not to add the device swap as primary swap, the system will use the NFS swap entry during the initial part of the boot process and then use the device swap (it will usually be priority 1 which is higher than the NFS swap at priority 5).

Cluster-wide Tasks

- Question:** Is a resource considered to be a cluster-wide resource if it has been added to every system in a cluster via separate **Add** tasks (that is, the **Manage Cluster-Wide ...** option was *not* used)?
- Answer:** No, adding a resource to each system in a cluster one at a time does *not* create a cluster-wide resource.
- Even though the result is the same as if you had done a cluster-wide task, SAM cannot know with certainty that you want the resource treated as a cluster-wide resource. But you can still use the **Manage Cluster-Wide ...** action to let SAM know that you want to manage the resource cluster-wide from now on.
- Question:** If a cluster-wide resource is removed from a system, how can it be added back later?
- Answer:** The best approach is to run SAM on a system where the resource is unconfigured, select the resource, and select **Add Unconfigured**. This gives you the option of adding the cluster-wide resource to just the local system, or to all systems in the cluster where the resource is not configured.
- Question:** I have a cluster-wide resource, but I have made some local changes to the configuration of that resource on some of the systems in the cluster. When I add a new system to the cluster, how is the resource configured on that system?
- Answer:** Local tasks performed on a cluster-wide resource do not affect the cluster-wide configuration of that resource. The resource will be configured on the new client the way it was added, or last modified *cluster-wide*.

B-12 NFS Diskless Questions and Answers

For example, suppose you do a cluster-wide add of a printer with the fence priority set to 1. Then you do a cluster-wide task to modify the fence priority to 4. At that point, if a new system is added to the cluster, the printer is configured on that system with a fence priority of 4.

Question: If I have removed a cluster-wide resource from a system in a cluster, and then I add that resource back to that system but with a different local identifier (that is, a different local mount point for a file system, or a different local name for a printer), will future cluster-wide tasks affect this system?

Answer: No, SAM treats the resource on that system as a different resource.

A cluster-wide resource is uniquely identified by two values: the identifier by which the resource is referred to on systems where it is a remote resource (for example, the local mount point of an NFS file system) and the location of the resource (for example, `hostname:mountpoint` of an exported file system).

So if you have configured a resource on some systems in a cluster with one local identifier, and on other systems in the cluster with a different local identifier, SAM can't assume that they are the same resource.

Question: What do I do if I have a relatively large cluster (say, 100 systems) and I only want about half of them to have access to a resource?

Answer: This is a case where SAM is not going to be much more help than if you had a large collection of stand-alone systems and you wanted some of them to have access to a resource. SAM helps you manage all of the systems in a cluster consistently, with some flexibility to allow for exceptions, but does not help you manage subsets of a cluster.

Question: Do I have to add and manage cluster-wide resources by running SAM on the cluster server?

Answer: No, in fact with NFS Diskless there is much less of a distinction between a cluster server and cluster clients.

In general, you can run SAM on any system in a cluster to manage cluster-wide resources. You do have to run SAM on the system that a resource is attached to in order to do a local add or remove of that resource.

For example, if you attach a printer to a cluster client named **zorro**, then you must run SAM on **zorro** to configure the printer on **zorro**. To make the printer a cluster-wide resource you can either:

- Use the “Manage Cluster-Wide” option when doing the local add;
or
- Run SAM and do a cluster-wide remote add while on one of the other members of the cluster;
or
- Change the cluster-wide state of the printer while running on the system that the printer is attached to.

File Systems

Question: When a physical file system is made into a cluster-wide resource, what does SAM do?

Answer: SAM uses the automounter to access the file system on the other systems in the cluster. The steps are:

- SAM modifies `/etc/exports` to allow the other systems access to the file system.
- SAM creates an entry in an automounter direct map that is managed by SAM (`/etc/auto_cluster`).

Users and Groups

Question: What if I want to use NIS to manage user/group data?

Answer: SAM cluster configuration provides one method of sharing user/group data, home directories and mailboxes among all of the members of a cluster.

There are certainly other methods of accomplishing the same goals. One example is NIS; another possible configuration is to have a mail server or home directory server that is different from the cluster server. The way to set up one of these alternate sharing mechanisms is to select **Private** in SAM as the policy for users and groups. Selecting **Private** means that SAM creates the same configuration on each member of a cluster as if you did a “cold-install” of a standalone system. You can then “manually” set up the alternative sharing mechanism (that is, not using SAM).

Question: What if I want to change policies after I have created a cluster?

Answer: SAM does not allow you change policies after the first client has been added to a cluster. The only way to change policies using SAM is to delete all clients from a cluster, pick different policies when re-adding the first client, and then re-add the other clients.

It is possible to modify the cluster configuration manually (not using SAM) to change a policy, but if you then use SAM to add more clients to the cluster, SAM adds the new clients in accordance with the original policy, ignoring your manual modifications.

Question: How do I modify the `passwd` file when a shared user/group policy is in force?

Answer: You need to be careful in this case.

A program that unlinks one of these files in the process of changing it breaks the sharing mechanisms set up by SAM. Use one of following methods:

- Use a supported command (such as `passwd(1)`, `vipw(1M)`, `chsh(1)` or `chfn(1)`).
- If you need to unlink the file, perform the operation on `/etc/share/passwd`, not `/etc/passwd` (and `/etc/share/group` rather than `/etc/group`).

For example:

```
cp /etc/share/passwd /etc/share/passwd.new
```

```
vi /etc/share/passwd.new
```

```
mv /etc/share/passwd.new /etc/share/passwd
```

- Use code such as the fragment that follows to modify the password file programmatically.

Modifying the Password File Programmatically

The code fragment on the next page looks for a user in the password file with the user name found in the variable `login_name`, and replaces the home directory for that user with the value found in the variable `new_directory`. The algorithm makes a copy of the password file before modifying it, then replaces the existing password file with the modified file in an atomic operation.


```

strcpy(passwd_file, "/etc/passwd");

/* follow possible symbolic links to get actual password file */
while(1) {
if (lstat(passwd_file, &file_info) != 0)
ERROR
if ((file_info.st_mode & S_IFMT) == S_IFREG)
break;
if ((file_info.st_mode & S_IFMT) != S_IFLNK)
ERROR
if (readlink(passwd_file, follow_link, 100) < 0)
ERROR
strcpy(passwd_file, follow_link, rc);
passwd_file[rc] = '\0';
}

/* block simultaneous access attempts at password file */
lckpwwd();

/* open temporary password file */
strcpy(temp_pwd, passwd_file); strcat(temp_pwd, ".tmp");
tf = fopen(temp_pwd, "w");
if (tf == NULL)
ERROR

/* copy existing passwd file to temporary file, modifying desired entry */
found = 0; setpwent();
while((pwd = getpwent()) != NULL) {
if (strcmp(pwd->pw_name, login_name) == 0) {
found = 1;
strcpy(pwd->pw_dir, new_directory);
}
putpwent(pwd, tf);
}
endpwent(); fsync(fileno(tf)); fclose(tf);

if (!found)
ERROR

/* replace existing passwd file with modified file */
if (rename(temp_pwd, passwd_file) < 0)
ERROR

/* unlock password file */
ulckpwwd();

```

