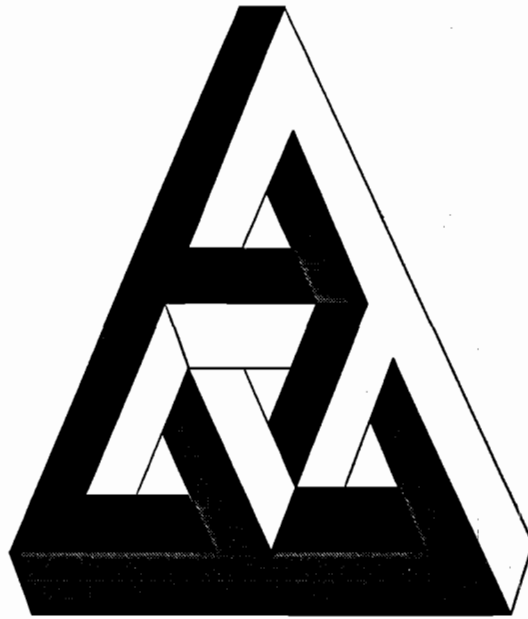




HP-UX

10.0

DELTA

HP-UX 10.0 Specialist Delta Training

Version 1.1.1

2

10/1/1914

10/1/1914

1

10/1/1914

2

2



HP-UX 10.0

Specialist

Delta Training

Hewlett-Packard Company
Operating Systems Software Division
19111 Pruneridge Avenue
Cupertino, CA 94091-0795

NOTICE

The information in this document is subject to change without notice. Hewlett-Packard Company (HP) provides this material "as is" and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. HP shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract or other legal theory.

Some states do not allow the exclusion of implied warranties or the limitation or exclusion of liability for incidental or consequential damages, so the above limitation and exclusions may not apply to you. This warranty gives you specific legal rights. You may have other rights, which vary from state to state.

This document contains proprietary information which is protected by copyright. All rights are reserved by HP. No part of this document may be photocopied, reproduced, or translated to another program language without prior, expressed, written consent of HP.

Printing history

Version 1.0

Printed 10/26/94

Version 1.1

Printed 1/4/95

Version 1.1.1

Printed 1/16/94

HP-UX 10.0 Specialist
Delta Training Course Overview
xv

General course information	xvi
Course objectives.....	xvii
Course agenda and schedule	xviii
Course topics and length of presentations.....	xix
Descriptions of course modules	xx
Typographical conventions used in this course book ..	xxiii

Introduction to HP-UX 10.0
1

HP's operating environment strategy and HP-UX 10.0...	2
Major new features of HP-UX 10.0	3
Old features replaced by new and improved features.....	4
HP-UX 10.0 standards compliance	5
Supported file system types.....	7
Supported SPU's on HP-UX 10.0	8
Desktop HP-UX (S700 only)	9
What is in Desktop HP-UX?	10
WSIO (S700) vs. SIO (S800) software functionality.....	11
Summary and review	12
For more information.....	13

Pre-Upgrade Analysis Tools
15

Motivations for providing pre-upgrade analysis tools....	16
Features and benefits of pre-upgrade analysis tools	17
New terms and technologies.....	18
Pre-upgrade analysis tools	19
Pre-upgrade tasks	20
Running prepare	21
Overview – Running prepare	22
Step 1 – Running prepare -g.....	23
The prepare “.g” files.....	24
Step 2 – Running prepare -q.....	25
Step 3 – Running prepare -S, -m.....	26
The prepare -m option.....	27
Modifying prepare's behavior: .preparerc.....	28

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Using analyzer	29
Tips and tricks	30
Summary and review	31
For more information	32
Written exercises	33
Lab exercises	34
 Installing HP-UX 10.0	 39
Motivations for performing a cold install	40
Requirements	41
Ingredients for a cold install	42
Overview of installation steps	43
Step 1: Backup, copy, or print	44
Step 2: Start installation	45
Step 3: Configure system & load software	46
Step 4: Complete installation	47
Step 5: post-installation configuration	48
Troubleshooting and recovery	49
For more information	50
Lab exercises	51
 File System Layout	 55
Motivation to change file system layout	56
Features and benefits of	
HP-UX 10.0 File System Layout	57
File system layout overview	58
Directory tree	59
Major new directories in 10.x	60
New names for old directories	62
Application directories	64
Path variables	65
Overview of impacts	66
Transition links	67
Transition link utilities and database	68
Tools to locate files	69
Using fnlookup	70

Using classic tools—a quick review	71
Summary and review	72
For more information	73
Written exercises	74
Lab exercises	76
Additional notes for “File System Layout” module	77

User Environment 79

Motivations for user environment changes	80
Features and benefits of user environment changes	81
Default shell is OSF POSIX shell	82
Some major POSIX shell features not in Korn shell	83
Modified user-level commands	84
Changes to audio	89
The dtterm terminal emulator	90
The dtterm window options	91
Tips and tricks	92
Summary and review	93
For more information	94
Lab exercises	95
Additional notes for “User Environment” module	97
Other modified or new commands from Section 1	99

System Startup 101

Motivations for changing the model	102
Features and benefits of new model	103
9.0x vs. 10.0 scripts	104
Startup/shutdown model overview	105
Example execution script: cron	106
System startup (shutdown) checklist	107
Running execution scripts: /sbin/rc	108
The /sbin/rc*.d directories	109
Link files—“S”tart and “K”ill links	110
Start and kill link naming	111
Script names and locations	112
Example configuration variable script	113

Modifying configuration variable scripts.....	114
Run level definitions.....	115
Example startup sequence	116
Adding startup/shutdown scripts.....	117
Writing execution scripts	118
Creating start and kill links.....	119
Writing config variable scripts.....	120
Summary and review.....	121
For more information	122
Written exercises	123
Written exercises, continued.....	124
Written exercises, continued.....	125
Lab exercises.....	126
 Introduction to Software Distributor	 127
Motivations for changing software distribution tools...	128
Features and benefits of Software Distributor	129
SD-UX vs. HP-UX 9.0x tools	130
History of SD-UX.....	131
SD-UX system overview and terminology	132
SD-UX daemon and agent.....	133
SD-UX software objects.....	134
SD-UX input files.....	135
Software states.....	136
Control scripts.....	137
Installed Products Database (IPD) and depot catalog..	138
SD-UX commands listing	139
The swinstall command syntax.....	140
Syntax for software_selections.....	141
Example of SD-UX GUI: Software Selection Window ..	142
Listing information: swlist.....	143
Listing information about software: swlist	144
Installing software: swinstall.....	145
Compatibilities and dependencies	146
Product location and multiple versions	147
Kernel rebuilding and rebooting.....	148
Using the swinstall GUI: Phases of installation.....	149

Using the swinstall GUI: Results of analysis	150
Control scripts used by swinstall	151
Verifying software: swverify	152
Verification process: Analysis phase	153
Removing software products: swremove	154
Control scripts used by swremove	155
Configuring software: swconfig	156
Configuration process: Analysis phase	157
Copying a depot: swcopy	158
Registering depots: swreg	159
Controlling access with ACLs	160
Example ACL: for root	161
ACL levels	162
ACL entries	163
ACL permissions	164
Controlling access using swacl	165
Tips and tricks	166
Summary and review	167
Co-existence with HP-UX 9.0	168
For more information	169
Written exercises	170
Lab exercises	172
Additional notes for "Introduction to SD-UX" module ..	179
SD-UX commands' features comparison	181
Example of using swlist: Listing media attributes	182
Examples of using swacl	183
Permissions for using swinstall	184

Software Distributor Packaging 185

Software Distributor (SD-UX) package	186
Product Specification File (PSF)	187
PSF attributes for all objects	188
PSF: Vendor specification example	189
PSF: Product specification example	190
PSF: Subproduct specification example	191
PSF: Fileset specification example	192
PSF: File and directory specifications	193

PSF: Permissions specifications.....	194
PSF: Control script attributes.....	195
Control scripts "DON'T's"	196
Control scripts "DO's"	197
Example of control script: sample configure	198
Phases of packaging operation.....	199
Using swpackage command	200
Modifying an existing product: swmodify.....	201
Converting fpkg to swpackage format	202
Tips and tricks	203
Summary and review.....	204
For more information	205
Lab exercises	206
Additional notes for "SD-UX Packaging" module	211
Environment variables for use by control scripts	212
Examples of using swmodify	213
ACL permission requirements to package	214
 S700 and S800 I/O Convergence	 215
Motivations for convergence.....	216
Features and benefits	217
Overview of impacts	218
I/O features not in HP-UX 10.0.....	219
I/O Architecture in HP-UX 9.0x.....	220
HP-UX 10.0 converged I/O system.....	221
General I/O system (GIO).....	222
Context Dependent I/O	223
Example – S700 I/O Tree	224
Example – K-Series I/O Tree	225
Admin issues – special file changes	226
Device file name convention.....	227
Example – Device Mapping.....	228
Example – Tape device naming	229
Recognized tape devices	230
Example – Disk device naming.....	231
Example – Serial device naming.....	232
Overview – I/O utilities	233

System generation convergence	234
Converged ioinit features	235
Converged ioscan features	236
Converged *sf commands	237
Converged disk layout	238
Hard partition convergence	239
Compatibility Pseudo Driver (CPD).....	240
Developers' issues – Major Number assignment.....	241
New dev_t format.....	242
Required changes to drivers	243
Coherent I/O	244
Coherent I/O features	245
Summary and review	246
For more information.....	247
Written exercises.....	248
Lab exercises	249

Logical Volume Manager (LVM) 253

Motivations for LVM changes	254
Features and benefits	255
Partitioning vs. Logical Volumes	256
History of LVM-related features	257
Overview of new features	258
LVM now supported on S700.....	259
Automated configuration backups	260
Physical Volume (PV) links	261
PV link commands	262
Viewing PV links.....	263
V.4 file system layout for files and commands	264
Commands for physical volumes.....	265
Commands for volume groups and logical volumes	266
Atomic multiple lvsplit	267
Multiple computers sharing disks.....	268
Read-only volume groups.....	269
Example - Using lvsplit and read-only volume group..	270
Old features preserved.....	271
Old features replaced	272

Migrating from SDS to LVM stripes.....	273
Mirroring and striping (<i>post- 10.0</i>)	274
SDS to LVM migration tools	275
Migrating from DataPair to LVM.....	276
Migrating from old non-LVM disks	277
Tips and tricks	278
For more information	279
Written exercises	280
Lab exercises	281
 Journalled File System (JFS)	 285
Motivations for providing JFS	286
Base JFS vs. OnLine JFS features	287
Fast file system recovery	288
The intent log	289
Extents and extent-based allocation.....	290
JFS layout	291
The allocation unit (AU).....	292
Components of an AU	293
Old commands that know about JFS.....	294
Commands new with JFS.....	295
Performance and tuning	296
New mount options	297
OnLine JFS: On-line system administration	298
OnLine JFS: On-line backup and snapshots.....	299
OnLine JFS: New commands.....	300
On-line JFS: new mount options.....	301
Things to know about JFS.....	302
Tips and tricks	303
Summary and review.....	304
For more information	305
Written exercises	306

Kernel Generation**307**

Motivations for changing	308
Features and benefits	309
Why generate a new kernel?	310
Ways to rebuild the kernel	311
SAM's capabilities	312
SAM's kernel configuration area	313
Manually generating a kernel	314
Main ingredients in kernel gen	315
Driver changes	316
LAN driver changes	317
Format of the system file	318
New tunable kernel parameters	319
Changing the console device	320
Changing the root device	321
Setting the dump device	322
Setting swap	323
Steps to add a non-HP driver	324
Summary	325
Written exercises	326
Lab exercises	327

Diagnostics**331**

Motivations for changes	332
Features and benefits	333
HP-UX 9.0x vs... ..	334
...HP-UX 10.0	335
Memory Page Deallocation	336
Spotting failed memory	337
Support media at 10.0	338
Changes to offline diagnostics	339
Changes to online diagnostics	340
Tips and tricks	341
For more information	342
Written exercises	343
Lab exercises	344

SAM and Administrative Commands	347
Motivations for changing SAM and admin commands .	348
Features and benefits	349
SAM Functional Area Launcher (FAL)	350
SAM FAL: ASCII terminal interface	351
"Accounts for Users and Groups" changes	352
"Backup and Recovery" changes	353
"Disks and File Systems" changes	354
"Kernel Configuration" changes.....	355
"Peripheral Devices" changes.....	356
"Printers/Plotters" screen.....	357
"Process Management" changes.....	358
"Routine Tasks" changes	359
"Time": new functional area	360
Overview of other major changes to SAM	361
SAM Log Viewer	362
SAM log file viewer filtering options	363
Restricted SAM Builder overview.....	364
Using Restricted SAM Builder.....	365
Custom SAM utilities: adding custom groups.....	366
Custom SAM utilities: adding custom applications.....	367
New administrative commands	368
Multiple file systems support-related changes	370
Changes to miscellaneous admin commands	372
Changes to newfs command	374
System accounting modification	375
Summary and review.....	376
For more information	377
Lab exercises	378
Additional notes for "SAM and Admin Commands"	385
The Object-Action Manager (ObAM)	386

The Virtual Memory System	387
Features and benefits	388
Overview – Converged features	389
Virtual address space changes	390
Memory mapped files	391
Programming calls for MMFs	392
The <code>vfork(2)</code> system call	393
EXEC_MAGIC executable	394
Changes to VM Model	395
Pseudo swap	396
Paging system changes	397
Changes to <code>vhand</code>	398
Changes to the swapper (<code>sched</code>)	399
<code>serialize</code> and <code>renice</code>	400
Summary and review	401
 Process Resource Manager (PRM)	 403
What is PRM?	404
Why use PRM?	405
Standard HP-UX scheduler	406
Introduction to the PRM scheduler	407
The standard scheduler compared with PRM	408
The carousel	409
Installing PRM	410
Questions to plan the PRM configuration	411
Steps to implement the configuration	412
The configuration file <code>/etc/prmconf</code>	413
The simplest configuration file	414
Configuration file utilities	415
PRM states	416
Primary administration: <code>prmconfig</code> command	417
Dynamic changes and fine-tuning the configuration ...	418
The <code>prmmove</code> command	419
HP-UX commands that know about PRM	420
Monitoring CPU usage	421
The <code>prmmonitor</code> command	422

Performance impact.....	423
Lab exercises	424
Trusted Systems	429
Motivations for changing trusted systems features.....	430
Features and benefits of new trusted systems features	431
Protected password file changes	432
Password generation	433
Password screening	434
Password aging	435
Time-based access controls.....	436
Account locking and login control	437
Location-based access controls.....	438
New and obsolete developer calls.....	439
Tips and tricks	440
Summary and review.....	441
For more information	442
Written exercises	443
Course Review	445
Summary and review of changes to HP-UX.....	446
What else has changed at HP-UX 10.0?	448
Any questions?	450

HP-UX 10.0

Specialist

Delta Training

Course Overview

General course information

Audience

Response Center Engineers, Expert Center Engineers,
Customer Engineers, PSO Technical Consultants

Synopsis

"HP-UX 10.0 Specialist Delta Training" presents a detailed look at many of the changes in the HP-UX operating system between HP-UX 9.0x and HP-UX 10.0. Subjects presented include supported standards and systems, pre-upgrade analysis tools, installation, file system layout, user environment, system startup, Series 700 and 800 I/O convergence, Software Distributor (SD-UX), Logical Volume Manager (LVM), kernel generation, diagnostics, SAM and administrative commands, virtual memory, the Process Resource Manager (PRM), and trusted systems. Additionally, the journaled file system (JFS), available at HP-UX 10.01, will be discussed. Those changes not covered are listed in the "Review" module at the end of this course book.

Prerequisites

- Knowledge of HP-UX 9.0x operating system concepts and principles
 - *How HP-UX Works: Concepts for the System Administrator* provides this prerequisite information
- Knowledge of HP-UX 9.0x system administration
 - *System Administration Tasks* (for the HP 9000 Series 700 and 800 systems) provides this prerequisite information

Course Length

1 week (5 days)

Each day 8 AM - 6 PM

Course Format

Lecture, written exercises and approximately 50% lab

Course objectives

1. Provide initial exposure to the course participant on the HP-UX 10.0 operating system by describing in detail the changes to HP-UX that occurred between HP-UX 9.0x and HP-UX 10.0.
2. Prepare the course participant for providing technical support to HP-UX 10.0 customers, including users, system administrators and programmers.
3. Allow the course participant to apply and reinforce concepts learned in the class lectures by accomplishing labs and completing written exercises.

Course agenda and schedule

This may vary slightly with each class

	Mon	Tues	Weds	Thurs	Fri
AM	Overview, Introduction Pre-upgrade analysis	User env. System startup	I/O converg. Logical Volume Manager	Kernel generation Diagnostics	Proc. Res. Mgr. Trusted Sys Review
PM	Installation Filesystem layout	Software Distributor SD-UX Packaging	LVM (cont'd.) JFS	SAM & Admin commands Virtual memory	Open Lab

Monday - Friday schedule: 8 am - 6 pm

1 hour lunch break each day

Friday afternoon is reserved for open lab time

Course topics and course times

Course topic	Length (approximate)
Overview/Introduction	2 hours
Pre-upgrade analysis	2 hours
Installation	2 hours
Filesystem layout	2 hours
User environment	.5 hour
System startup	2.5 hours
S700 and S800 I/O convergence	3 hours
Software Distributor introduction	5 hours
Software Distributor packaging	2 hours
Logical Volume Manager (LVM)	3 hours
Journalled file system (JFS)	2 hours
Kernel generation	3 hours
Diagnostics	2 hours
SAM and administrative commands	3.5 hours
Virtual memory	1 hours
Process Resource Manager (PRM)	2.5 hours
Trusted systems	1 hour
Course review and wrap-up	1 hour

Descriptions of course modules

Module 1 - Introduction to HP-UX 10.0

Provides an overview of changes to HP-UX, including standards conformance, major new and discontinued features, and supported filesystems and hardware.

Module 2 - Pre-Upgrade Analysis

Describes the tools and process used to perform an analysis of an HP-UX 9.0x system before upgrading. (Upgrade tools are not available at 10.0, but will be available post-10.0.)

Module 3 - Installation

Shows how to cold-install HP-UX 10.0. Explores requirements and media options for installation. Troubleshoots common problems with installation.

Module 4 - File System Layout

Explains the motivation for revising the file system layout. Describes in detail the changes to the file system, and provides instructions on resolving pathnames.

Module 5 - User Environment

Lists modifications to commands commonly used by users. Describes the POSIX shell features and functionality. Provides overview of changes to VUE.

Module 6 - System Startup

Explains features and benefits of new startup scripts model. Describes layout and contents of directories and startup scripts. Discusses how to customize and invoke startup scripts.

Module 7 - Introduction to Software Distributor

Provides a detailed description of the components of the Software Distributor (SD-UX) system, including commands necessary to list, install, verify, configure and remove software. Additionally, covers access control.

Module 8 - Software Distributor Packaging

Describes the Software Distributor (SD-UX) packaging process, including scripts to manage the process and the structure of the file specifying product information.

Module 9 - S700 and S800 I/O Convergence

Describes the characteristics of I/O convergence for the Series 700 and 800, including architecture, device special file changes, disk layout and utilities for administering I/O devices.

Module 10 - Logical Volume Manager

Describes modified and new features of Logical Volume Manager (LVM), including its replacement of DataPair and Software Disk Striping (SDS).

Module 11 - Journaled File System

Discusses the journaled file system (JFS), the new file system type available at HP-UX 10.01. Features presented include fast recovery, on-line backup, resizing and reorganization, and new and modified commands.

Module 12 - Kernel Generation

Explains the process for kernel generation and kernel tuning. Describes newly supported kernel configurations, including small disk and small memory configurations.

Module 13 - Diagnostics

Discusses memory page deallocation, diagnostics licensing, and the new off-line and on-line diagnostics usage and features at 10.0.

Module 14 - SAM and Administrative Commands

Describes SAM's modified user interface, functionality and logging. Provides an introduction to new features. Additionally, covers various new and modified administrative commands.

Module 15 - Virtual Memory

Describes changes to demand paging, process aging and page stealing, process deactivation and reactivation.

Discusses the new memory mapped files.

Module 16 - Process Resource Manager

Describes features of Process Resource Manager (PRM), a purchasable product that replaces the Fair Share Scheduler (FSS).

Module 17 - Trusted Systems

Describes changes to commercial security features, including new password options and control of terminal access.

Module 18 - Course Review

Provides a review of the entire course, enabling course participants to check their learning, clarify any remaining issues, and determine the next step for learning more about HP-UX.

Typographical conventions used

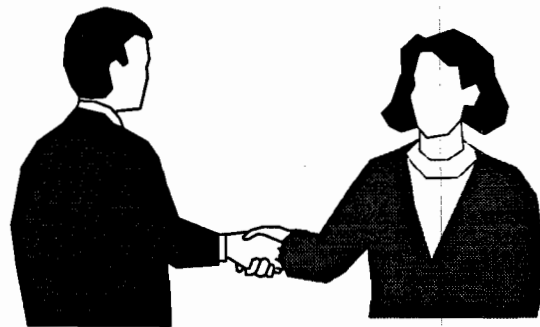
This book uses the following typographical conventions.

If text looks like this	It indicates
<code>computer text (courier)</code>	text displayed by the computer system or text that should be entered at the keyboard
<i>menu item (italics)</i>	an on-screen menu item or option

- *Italics* may also be used to indicate a new term is being introduced.
- Occasionally, *italics* or **bolding** may be used for emphasis.
- In general, the context in which the word is used will indicate why the type looks different.

This page intentionally left blank.

Introduction to HP-UX 10.0



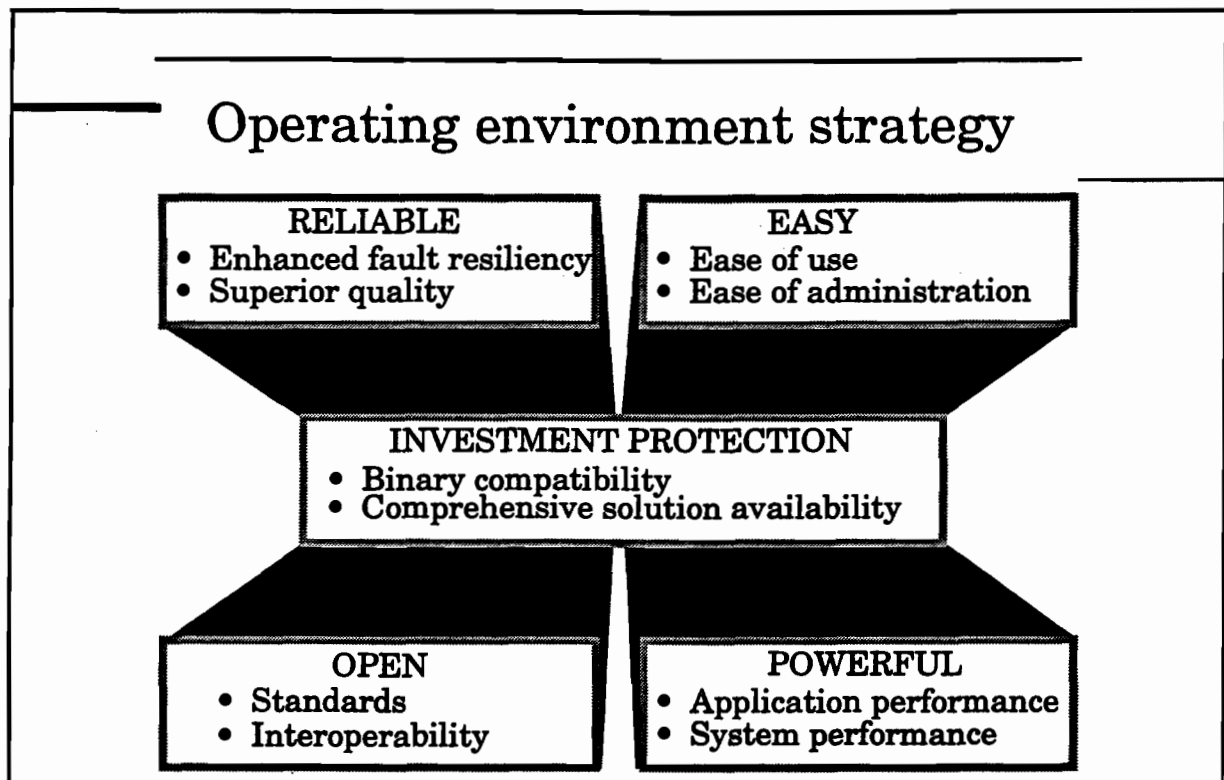
Module goals:

The purpose of this module is to introduce HP-UX 10.0 features and characteristics.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe HP's operating environment strategy.
2. List major new features of HP-UX 10.0, bundled and unbundled.
3. List features which are discontinued, and their replacements.
4. List the computer industry standards to which this release conforms.
5. Compare and contrast supported filesystem types.
6. Identify SPUs supported by HP-UX 10.0.
7. Describe Desktop HP-UX, including what it does and does not contain.
8. Identify functionality differences remaining between S700 and S800.



Notes

- HP's operating environment strategy is designed to meet customers' needs and continue HP's position as a high-quality industry leader.
- HP-UX 10.0 was developed to meet the requirements of HP's operating environment strategy.

Major new features

Bundled

- SVR4 file system layout
- S700 and S800 I/O convergence
- Logical Volume Manager on S700
- Software Distributor (SD-UX) for managing software installation, updating and packaging
- Desktop HP-UX and Lite HP-UX (S700 only)
- Support for new SPUs

Unbundled (purchasable)

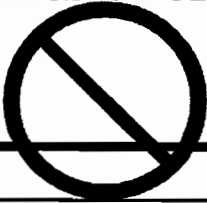
- HP Process Resource Manager (PRM)
- MirrorDisk/UX
- MC/ServiceGuard



Notes

- Additional features include improved performance, new trusted systems and security features and bundled SharedPrint.

Replacements of old features



Still Available?	Feature	Replacement
NO	DataPair/UX	MirrorDisk/UX (with LVM)
NO	Software disk striping on S700	Logical Volume Manager now includes feature
YES	Hard partitions on S800	Logical volumes
NO	DUI (/etc/update, etc.)	Software Distributor (SD-UX)
YES	Bourne shell	POSIX shell
NO	DUX	<i>Post-10.0: NFS Diskless</i>
YES	Switchover/UX	MC/ServiceGuard

Notes

- Some of these discontinued features are available on HP-UX 10.0, but it is recommended that customers use the newer product.
 - In many cases, these discontinued features will not be available in the next release.
 - The new features are industry-standard and/or more flexible and powerful.

HP-UX 10.0 standards compliance

- Continued convergence on industry standards
- Provides a more open environment
- Continues HP's commitment to be at the forefront of UNIX industry standards conformance
- Brings additional COSE compliance



Notes

- By complying with standards, either *de facto* or those officially approved by an accepted representative organization, HP-UX will allow customers to
 - work more easily within a multi-vendor environment
 - transition more easily from another UNIX platform to HP-UX

HP-UX 10.0 standards compliance

- SVR4 file system layout
- XPG4
- PEX (3D graphics)
- SVID3 base and kernel extensions
- OSF AES (on S800)
- POSIX 1003.4 (real time) (partial)
- POSIX 1003.1c (threads) (draft 4 user threads)
- 4 Byte EUC (Asian language support)



Notes

- POSIX 1003.4 partial real time
 - Includes synchronized I/O, timers and priority scheduling
- POSIX 1003.1c - user space threads
 - Currently available with DCE

Supported file system types

	HFS	NFS	DFS	JFS
Real disk-based file system	Yes	No	No	Yes
Remote file system	No	Yes	Yes	No
Exportable by NFS	Yes	No	Yes	Yes
Exportable by DFS	Yes	No	N/A	Yes
Other names	UFS	--	--	VxFS
Developer	UCB	Sun	Transarc	Veritas
Shipped by HP as part of:	9.0/10.0	9.0/10.0	9.0/10.x	10.01

- Additionally, LOFS and CDFS are supported at 10.0.

Notes

- HP-UX 10.0 has support for multiple local file systems.
- LOFS is the new loopback file system.
 - This is a “fake file system”. A file system can be mounted in more than one location simultaneously. The effect is similar to that of a symbolic link, but the usual mount options are available, and there are no issues with the meaning of “..”.
 - Recommended read-only!
 - Performance is better than local NFS mounted file systems.
- Previously existing file systems were updated to fit new scheme.
- JFS will be added in HP-UX 10.01.

Supported SPUs on HP-UX 10.0

Supported

- All S800s that run 9.0x
- S840, S822, S825*
- All S700s that run 9.0x
- Future J class**
- Future K class**

Not supported

- S808, S815
- S635SV, S645SV
- All 68k-based
SPUs (S300,S400)

* Machines with 8 MB memory cards in first slot will not boot. They will work if the first card is >8Mb, however.

** New systems available Q195.

Notes

- The K class machines (known internally as Kittyhawks) will be the newest additions to the S800 server family. They are scheduled for release first quarter, calendar year 1995.
- The J class machines (known internally as Skyhawks) will be the newest additions to the S700 workstation family. They are scheduled for release first quarter, calendar year 1995.
- More information on these future systems will be provided at a later date.
- S300 and S400 systems are no longer supported.

Desktop HP-UX (S700 only)

- For commercial client/server market
 - Those who buy thousands of seats to run mission-critical applications on the desktop
- Subset of filesets from standard 2-User HP-UX Runtime product for workstations

Lite HP-UX for Desktop HP-UX

- Light-weight kernel configuration exclusively bundled with Desktop HP-UX SD bundle
- Saves > 1.5 MB of system memory

HP-UX $\Rightarrow$ Desktop HP-UX

Notes

- Desktop HP-UX requires less memory and significantly less disk space.
- It provides a single-user environment for running applications.
- This will be available through installation with Software Distributor (SD-UX)
- It is not available on S800.
- There is no trivial way to identify (e.g., uname) Desktop HP-UX.
- *History:* Available at 9.03 on S700 as subset of Instant Ignition.
 - There are some differences between Desktop HP-UX at 9.0.3 and at 10.0, because 9.0.3 was a minor release so the filesets were not completely arranged as desired.
- Lite HP-UX provides scaled-down set of HP-UX kernel resources and enables a minimal (core) set of critical subsystems.
 - Discussed in more detail in “Kernel Generation” module.

What is in Desktop HP-UX?

In	Not in
HP-UX kernel	Starbase, PEX, LVM
Shared libraries	C headers, most archive libraries
Most commands	On-line man pages, HP-UX-HELP
X11/VUE 3.0	Keyshell
Internet services	Server support for DTC, network install, diskless or font service
NFS/NIS client	UUCP, SLIP, NFS/NIS server
DCE/NCS	Accounting, disk quotas, auditing

Notes

- The original idea was to have a desktop box with a small disk and a small root file system.
 - For example, a 260 MB disk with room left over after installation.

WSIO vs. SIO functionality

WSIO-only

- Network cold install
- EISA
- Bitmapped drivers
- iomap(7)
- scsi_ctl(7) (pass through driver)
- Starbase graphics
- HP-PHIGS and HP-PEX
- Desktop HP-UX supportability

SIO-only

- Fiber Link support
- Device powerfail
- NIO (HP-PB)

Notes

Summary and review

Feature	Replaces	HP op. env. strategy supported
Standards compliance	–	Openness, Investment protection
I/O convergence	–	Easy to use/administrate
Logical Volume Manager	S800 hard partitions	Easy to administrate, Openness
Software Distributor	update, etc.	Easy to administrate
HP Process Resource Manager	FSS	Power and performance
MirrorDisk/UX	DataPair/UX	Reliability
Support for new SPUs	Support for 68k	Power and performance
Desktop HP-UX/Lite HP-UX	–	Easy to use/administrate
MC/ServiceGuard	Switchover/UX	Reliability

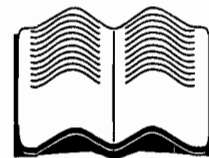
Notes

For more information



For more information

- General reference
 - HP-UX 10.0 Release Notes

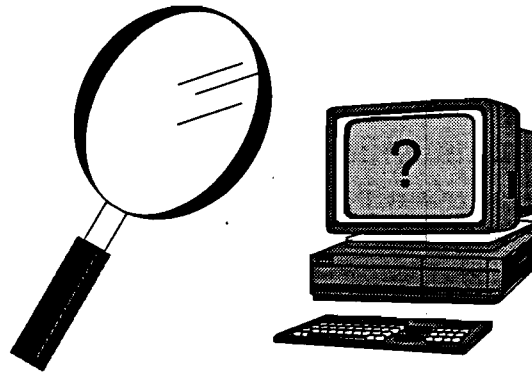


Notes

- The rest of this course covers new software features of HP-UX 10.0 in detail.

This page intentionally left blank.

Pre-Upgrade Analysis Tools



Module goals:

The purpose of this module is to describe the pre-upgrade analysis tools.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe motivations for and features and benefits of the pre-upgrade analysis tools.
2. Describe the purpose of each pre-upgrade analysis tool.
3. Run the pre-upgrade analysis tools and interpret their output.

Motivations

- Changes to commands, path names, and functionality require a comprehensive solution
- Three phases to upgrade
 - Pre-upgrade analysis
 - 9.0x -> 10.01 upgrade (*tools available post 10.0*)
 - Post-upgrade reconfiguration, verification and program modification (*tools avail. post 10.0*)
- Pre-upgrade analysis tools are available as part of HP-UX 10.0
 - Additionally, available as 9.U1 fileset
- *The upgrade and post-upgrade tools will come as part of HP-UX 10.01*

Notes

- HP-UX 10.0 represents a large change from 9.0x (file system layout, system start-up scripts, etc.), so it is easier to create a new, comprehensive scheme than to change the pre-existing one.
- Pre-upgrade analysis should be done long before the actual upgrade.
- These tools come with the HP-UX 10.0 media, and can be installed on HP-UX 9.0x systems using /etc/update.
 - By default, they install into /upgrade/bin directory on 9.0x.
- For HP-UX 10.0, these tools install in /opt/upgrade/bin directory.
- NOTE: The upgrade and post-upgrade tools are not available until post-10.0

Features and benefits

- These tools assist customers in determining impact of upgrading from HP-UX 9.0x to 10.01
- *Upgrade tools not available until post-10.0*
- They help customers to determine what files need to be modified and in what way
 - Not all situations can be anticipated
 - All changes should be manually inspected and re-certified
 - The tools never modify original files



Notes

- Customers' environments may be highly customized in ways that can not be anticipated by the tools developers. Therefore, the ultimate responsibility for the software analysis must lie with them.
- The tools will flag areas that require manual changes by the user.

New terms and technologies

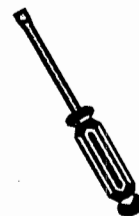
- Upgrade
 - The act of replacing 9.0x with 10.01
 - upgrade is also a program (replaces update)
 - *This is not available until post-10.0*
- TTDB, CMDS_DB, LIBS_DB
 - Locations (database files) where information used by pre-upgrade and upgrade tools is stored

Notes

- Don't call it a migration or reinstallation!
- There's also a DEV_DB file that shouldn't be modified.
- NOTE: Upgrade is not available until post-10.0.

Pre-upgrade analysis tools

Tool	Purpose
prepare	Creates and processes list of files requiring modification after upgrade
analyzer	Program that analyzes for required changes
fnlookup	Displays pre- and post-upgrade file pathnames



Notes

- Use for modifying scripts, makefiles, program sources and text files.
- Prior to the upgrade, use to assess the impact of the upgrade on system functionality, the option to make changes on a copy of the original file.
- Application developers who are upgrading their products to the 10.0 environment should also use these.
- The tools are covered in more detail in this module, except for fnlookup, which is discussed in the "File System Layout" module.

Pre-upgrade tasks

1. Scope software to understand quantity and type of changes required.
 - Use prepare (which calls analyzer)
2. Perform detailed analysis of all appropriate files.
 - Use analyzer and fnlookup
3. Make changes to copy of original file based on analysis data.
 - Use analyzer
4. Manually edit files based on recommendations.
5. Test changes for proper functionality and operation.

Notes

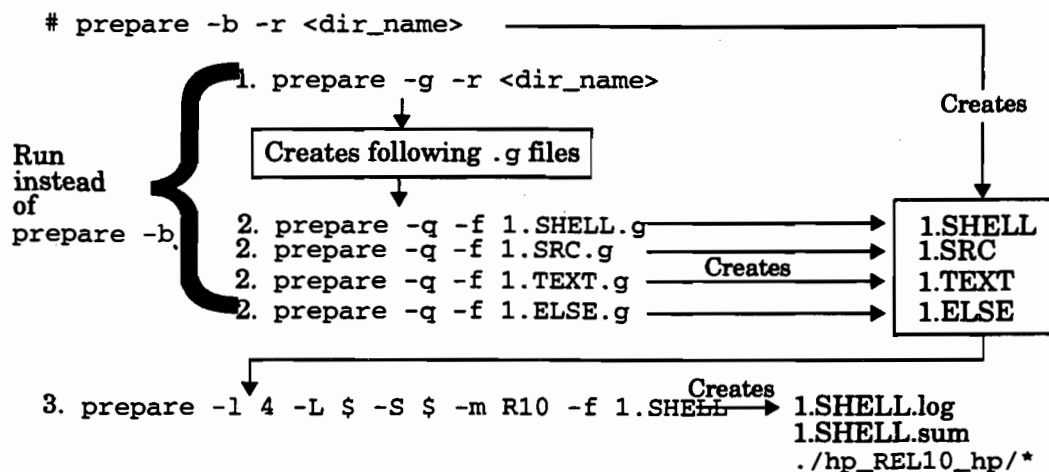
- This is the time to assess impact of upgrade!
 - File name and location changes
 - DUX cluster configuration data
 - Third-party applications
 - Interoperability with systems not upgraded

Running prepare

- Multi-function tool
 - Front-end for analyzer
- Checks large numbers of files and directories for required 10.0 upgrade changes
 - Also may be used to apply changes
- May be run in different ways:
 - Auto- (or Novice-) mode for ease-of-use
 - Expert-mode for control

Notes

Overview — Running prepare



Notes

- In the following slides the commands used in this process will be examined in more detail.

Step 1 — Running prepare -g

- Synopsis

```
prepare -g [-r] [-v|V] dir ... | -d dirlist
```
- Use **-d** and **-r** to specify directory tree to examine
 - System directories (/bin, /etc...) are **ignored** by default
- Customize behavior with **.preparerc** file
 - Must be in current working directory
 - Will be created if necessary
 - Controls files/directories examined by prepare
- Output is set of **“.g”** files
 - List of files potentially requiring modification
 - Files are **“SHELL”, “SRC”, “TEXT”, “ELSE”**

Notes

- Run prepare with no options to see usage.
- The first time it is run, it prompts to create a **“.preparerc”** file in the current directory. Allow this, unless another **.preparerc** file exists that is preferred to use.
- Optionally, first use **prepare -b** to get a thorough understanding of the process.

The prepare “.g” files

- Lists of file names segregated by type
 - SHELL Shell scripts and makefiles
 - SRC Program source files
 - TEXT Text files
 - ELSE Unrecognized files
 - SKIP Files skipped (binaries, etc.)
- Contains names of all files in tree of that type
 - Check/modify for “uninteresting” files
 - Lines commented-out will be ignored
- Examples
 - (if directory specified on command line) 1 .SHELL.g
 - (if -d users) users.SRC.g

Notes

Step 2 – Running prepare -q

- Synopsis

```
prepare -q [-v | V] [-pas] -f filelist
```

- Performs quick check of files in list
 - Files requiring at least one modification pass
- Also may be customized via .preparerc file
- Result is another list, with “.g” suffix stripped off
 - This list will be used during analysis
 - Modify by commenting-out lines

Notes

- Running prepare -b performs both steps 1 and 2.

Step 3 – Running prepare -S, -m

- Creates summary file containing specific changes required to files for upgrade and creates modified files in -m directory

- Synopsis

```
prepare [-V] [-l+log_lvl] [-S sum_file]  
[-m save_dir] file1
```

- Input file is that created from “-q” invocation
 - Wildcard, \$, allows substitution of file name
- Log levels go from 1 to 4; default is 2
 - Beware! Level 4 generates copious output
- Example:

```
prepare -V -l 4 -L $ -S $ -m . -f 1.SHELL
```

Notes

- The output files that will result from the example invocation will be named 1.SHELL.log and 1.SHELL.sum, respectively.

The prepare -m option

- Creates modified files in -m directory
 - Does not change originals!
- Use with the options as indicated on the previous page
- If do -m . , the save directory is ./hp_REL10_hp
- The analyzer automatically bypasses this directory so as not to analyze its own output
- Specify another directory if preferred
- Remember to prevent recursively analyzing this new directory by noting to bypass it in .preparerc

Notes

The .preparerc file

- Contains various options, many of which are user-modifiable, to control the running of prepare and analyzer
 - Which directories and files to bypass
 - For example, by pathname or owner
 - Whether or not to analyze comments
 - The maximum file size to bypass
 - Which options to pass to analyzer for each type of file analyzed
- Some “options” should not be modified, such as
 - System directories to bypass
 - Location of databases used by analysis tools

Notes

- A complete default .preparerc file is in the prepare man page.

Using analyzer

- Works on one file at a time, performing detailed analysis
- Synopsis

```
analyzer -[pascw][-l+log_level] [-m | -M mod_file ]  
in_file
```
- Modifications in -m or -M file
 - Original file never changed
- Logging details required modifications
 - Valid log levels are 0,1,2,3,4,+1,+2,+3,+4
- Obtains information about potential problems from CMDS_DB and LIB_DB
- Attempts to determine file type for analysis purposes, by file extension, for example

Notes

- Refer to analyzer man page for additional information.
- Messages from the CMDS_DB or LIB_DB are indicated in the log files by: C_### (e.g., C_162) or L_###, respectively
- To view the full text of messages, use logging level 4 or +4, or view the actual CMDS_DB or LIB_DB files

Tips and tricks

- Run prepare on a limited set of files
 - Keep things simple until you understand what it does
- Use the .preparerc to customize prepare and analyzer behavior
 - For example, to exclude or include files to be analyzed, or to place files on particular file type lists
- Use analyzer to analyze a single file



Notes

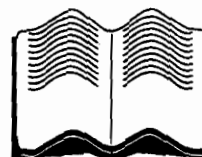
Summary and review

- Customers who plan to upgrade HP-UX 9.0x systems to HP-UX 10.01 should perform a pre-upgrade analysis on 9.0x systems far in advance of upgrading
- The prepare and analyzer tools are the pre-upgrade analysis tools that come with HP-UX 10.0
- These tools allow much flexibility for users
 - Use various command line options to control
 - Modify .preparerc file to control, if use prepare

Notes

For more information

- The man pages for each command
- The .preparerc file



Notes

A customer calls you up and asks your help in analyzing her 9.0x system in preparation for upgrading to 10.0. She wants to get started using analyzer. Help her out by providing the answers to the following questions:

1. What should be run instead of analyzer? Why?



2. Complete this table (to fax to her):

Command	Purpose/Description
<code>prepare -b -r <dir_name></code>	
<code>prepare -g -r <dir_name></code>	
<code>prepare -q -f <file_name></code>	
<code>prepare ... -S ...</code>	
<code>prepare -m ...</code>	

In this lab exercise, you perform a pre-upgrade analysis on a sample directory. Note that many of the changes the analysis indicates will be discussed in later modules. Your instructor will provide detailed instructions for loading the tools in your lab environment.

1. Enter: `cd /opt/deltaclass/labs/pre_upgrade`
List the files.
2. Run the `file` command on each, including the files in subdirectories.
Which files are English or ascii text?

Which files are commands text?

Which files are executables?

What are the file types of the remaining files?

3. Include `/upgrade/bin` in your `PATH`.
Include `/upgrade/share/man` in your `MANPATH`.
4. Run the `prepare` command (no options). Review the output. Note the suggested usage. Allow it to create the `.preparerc` file.
5. View the `.preparerc` file.

What is the max. size of files to be analyzed? _____

Which system directories are ignored by default?

What shell variable do you set to bypass specific files?

6. Enter: `prepare -g -V -r .` (current dir.)

What does `-V` option do? _____
(Refer to `prepare`'s man page).

What files did this create?

7. Which files are being skipped?
- _____

Which files are considered shell scripts?

Which files are considered text files?

Which files are considered "other"?

8. Enter: `prepare -q -V -f 1.SHELL.g`
Repeat for: `1.SRC.g`, `1.TEXT.g` and `1.ELSE.g`
What files do these commands create?
- _____
- _____

-
9. What are the differences, if any, in files listed in these files, versus files listed in the *.g files?

10. Enter:
prepare -V -l 1 -L \$ -S \$ -f 1.SHELL

What files are created?

View the summary file.

How many total warnings are there? _____

How many changed files are noted? _____

View the log file. What useful information is contained there?

11. Run the above command again.

What error do you get? _____

Copy these two files to new names, for later use:

```
cp 1.SHELL.sum 1.SHELL.sum1  
cp 1.SHELL.log 1.SHELL.log1
```

Look at the list of files near the end of .preparerc to see how prepare handles its own output files.

Look at the analyzer man page for details about logging levels.

12. Run the command again, but use log level +1:
prepare -V -l +1 -L \$ -S \$ -f 1.SHELL

View the new .log file.

Look at the last few lines of each analyzed file's "ANALYSIS SUMMARY".

What does this tell you about commands and pathnames for each analyzed file?

13. Copy the .sum and .log files to new names again:
cp 1.SHELL.sum 1.SHELL.sump1 (for +1)
cp 1.SHELL.log 1.SHELL.logpp1

14. Run the command again, but use log level 2:
prepare -V -l 2 -L \$ -S \$ -f 1.SHELL

View the new .log file.

The C_### (e.g., C_162) numbers refer to messages from the CMDS_DB.

Besides the C_### number, what information does this log file provide about the analyzed files (in addition to that provided in the 1.SHELL.log1 file)?

15. Copy the .sum and .log files to new names again:
cp 1.SHELL.sum 1.SHELL.sum2 (for 2)
cp 1.SHELL.log 1.SHELL.log2

-
16. Run the command again, but use log level 4:
`prepare -V -l 4 -L $ -S $ -f 1.SHELL`

View the new .log file. What does this tell you about the C_###, in addition to its number?

What does this tell you about lines that might have problems, in addition to the line number?

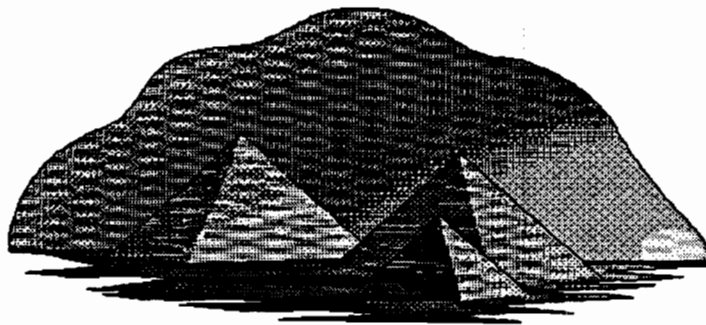
17. Optional: Run the prepare command with log level 3, and view the output, comparing with the other log levels.

18. For the 1.TEXT and 1.ELSE files, enter:
`prepare -V -l 1 -L $ -S $ -f 1.TEXT -m Rel10`
`prepare -V -l 1 -L $ -S $ -f 1.ELSE -m Rel10`

19. Change to the Rel10 directory. Look at the files in here. Compare files here with their original files. Were any changes made to the contents?
-

How does this help a customer to upgrade?

Installing HP-UX 10.0



Module goals:

This module explores the process of installing 10.0, and examines its options. By contrast to upgrading, installing completely destroys the old disk files and replaces them with all-new files.

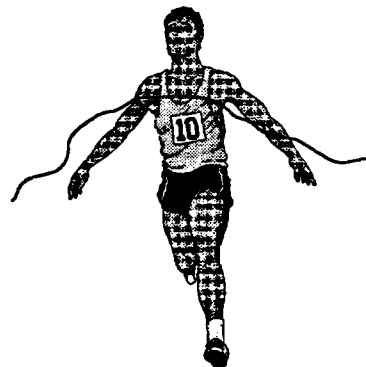
Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the media and options available at installation time.
2. Install HP-UX 10.0 on S700 or S800 from a tape or a network.

Why do a cold install?

- The only way to get 10.0 on existing hardware
 - New systems will have 10.0 by Instant Ignition
- To convert the root disk to LVM
- Existing 10.0 operating system was corrupted
- System disk was replaced



Notes

Requirements

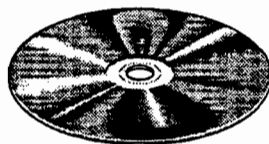
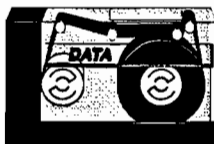
- Customer data must be backed up
 - Cold install wipes the existing disk clean
- The installation media itself
 - CD, DDS, networked software depot
- Hardware to read the media
 - Internal/external CD, DDS drive, network, ...
- S700 or S800
 - 16 Mb of memory
 - About 350 to 400 Mb of disk space
 - Graphics display or terminal as console

Notes

- To make a system disk image before installation, use the offline utility `dtutil`.
- Customers might also need a codeword if they install from a CD containing locked software.
- These requirements are very similar to those of 9.0x.

Ingredients

- Backup tapes of customer data
 - Cold install wipes the existing disk clean
- “Install” media
 - Usually DDS, CD, or network
- “HP-UX Core OS” media
 - Usually DDS, CD, or network

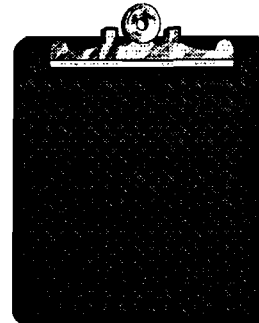


Notes

- If it's a bootable medium, what is its format?
- The difference in the software media between 9.0x and 10.0 is internal. Specifically, the HP-UX Core OS Software is no longer in Update format; it is in Software Distributor (SD-UX) format. You do not need to know this.
 - SD-UX is covered in much greater detail in two other modules of this class, so this module will not explain it in detail.
 - To sum up, SD-UX has a GUI. If you can push buttons and select items from a list, you can use SD-UX. You do not need to know the details of SD-UX to install HP-UX, and neither do customers.

Overview of steps

1. Back up or copy or print the important data
2. Start installation
3. Configure system & load software
4. Complete system set-up
5. Perform post-installation configuration

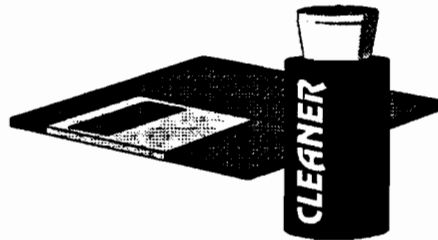


Notes

- The concepts behind these steps are the same as under 9.0x. The details are slightly different for 10.0 because the programs for installation are slightly different.
- The installation will stop if there is an error, so log files are not as necessary as they were under 9.0x. Warnings and remarks are still logged, and the log file name is announced at the beginning of the run.

Step 1: Backup, copy, or print

- Back up to tape with `fbackup`
- Copy to another system with `rcp` or `ftp` or `mailx`
- Make paper copies



Notes

- To make a system disk image before installation, use the offline utility `dtddutil`.
- Save anything that does not belong to the operating system, such as:
 - `/users/*`
 - File systems containing raw data, like a database
- Save anything that the administrator has edited, such as:
 - The `/etc/checklist` file; output from the `pvd`display, `vgdisplay`, and `lvdisplay` commands
 - The `/etc/resolv.conf` or `/etc/hosts` files; `/etc/netlinkrc`
 - `/.profile`, `.mailrc`, and other dot files for the super-user
 - `/etc/passwd` and `/etc/group` files, `/usr/lib/aliases`, `/usr/lib/sendmail.fc`, ...
 - Customized line printer models

Step 2: Start installation

- If re-installing an existing system, note the following before shutting down:
 - Hostname
 - IP address
 - Sub-network mask
 - Hostname and IP address of gateway
 - Hostname and IP of DNS & NIS servers (if used)
- Boot from alternate device or physical path
 - From CD, DDS tape, etc.
 - From a boot server over the network (700 only, and across at most 1 gateway)

Notes

- When booting off of the net, you will need to supply the proper boot string of the boot server on your local net or on a net at most one gateway distant from your local net. Use the “search” command from within the Boot ROM environment to determine the proper string.
- Note that once booted, installation of software on both S700 and S800 machines may be done from any machine that may be reached over the net. The one-hop restriction only applies to network booting of S700 machines.
- Once the boot process completes, you will see a menu that allows you to Install HP-UX or boot a Recovery Shell.

Step 3: Configure system & load

- Steps to perform prior to starting software load
 - Enable networking (if necessary)
 - Select Whole-System Configuration
 - Set Basic Configuration
 - Configure Disks and Filesystems
- Load Software
 - All available will be loaded if you choose not to interact with process
 - If you choose to interact, mark desired components and continue
 - Load “Core” media or specify install server’s IP
- After load, install builds new kernel & reboots

Notes

- Booting the installation media and configuring the system for installation will take about 10 to 15 minutes. Loading the software will typically take an hour or more, although this stage is non-interactive and you can safely go do something else.
- If you choose to interact with SD-UX, you can pick and choose which products are loaded from the depot.
- The installation process proceeds without user interaction once the files begin loading from the local media or install server.

Step 4: Complete installation

- Following reboot, `/sbin/set_parms` runs
- It asks about the following:
 - Hostname
 - Timezone
 - Time
 - Root password
 - IP address/subnet mask/gateway addr.
 - DNS
 - NIS
 - Miscellaneous (font server, etc.)

Notes

- After the installation is complete, the `/sbin/set_parms` script runs, and prompts for information regarding a number of system configuration items.
- Follow along with this script until it completes, after which the system will be ready for you to log in.

Step 5: post-install configuration

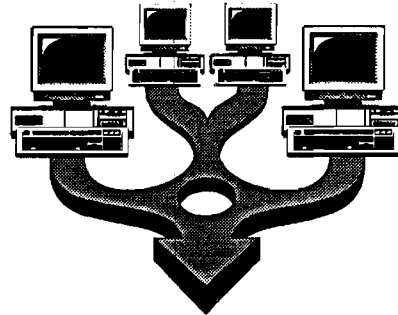
- Restore backups of user files
- Load any additional software
 - Diagnostics from the Support Media
 - Optional products if not on the software media
 - Databases or other third-party software
- Customize the system
 - Line printers
 - Terminals
 - Non-auto-configurable devices
 - Users
 - Other details as mentioned in the *System Administration Tasks* manual

Notes

- These steps are conceptually the same as under 9.0x, although the details are different:
 - Diagnostics are now on a different media (more on this in the “Diagnostics” module)
 - Third-party software, if written for 9.0x, may not run on 10.0 without modification or special precautions like compatibility links
 - Non-auto-configurable devices are installed differently (more in the “Kernel Generation” module)
 - The profile files for users will be different
 - A suggestion: Start with the new profile files, adding in old information; don’t blindly use the old profiles (\$PATH is new, ...)
- Note: The system does not need to run for half an hour after installation as it did at 9.0x. Diagnostics aren’t shipped with the system, and they install with less effort. The “Diagnostics” module covers this in more detail.

Troubleshooting and recovery

Problems	Solutions involve
Network problems	LLA correct? Correct boot table on server? Boot daemon running on server?
Incompatible media	Watch error messages carefully

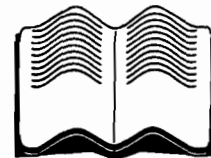


Notes

- In the lab, you'll get a chance to do some error recovery for problems caused by incompatible media.
- There will be a *Support Media User's Guide* at release. In the meantime, the Diagnostics lab provides the following summary of how to recover:
 1. Boot from media.
 2. Troubleshoot and repair the disk.
 3. Format the disk.
 4. Lay down a generic LIF volume.
 5. Build /sbin and copy in some archive commands, including recovery tools and a temporary kernel.
 6. Boot from the repaired disk.
 7. Use `frecover`, `tar`, or `cpio` to recover all data on the lost disk.

For more information

- *Installing and Upgrading to HP-UX 10.0*
- *Support Media User's Guide*



Notes

In this lab you will install HP-UX 10.0 on your system.

1. Before shutting your system down, gather some information:

Hostname:
IP address:
Subnet Mask:
DNS Server Name:
DNS Server IP Address:

2. Shut the system down and restart it. Interrupt the boot process so that you can interact with the Boot ROM.
3. Find out which systems will support the network installation process:

```
BOOT_ADMIN> search lan install
```

(Users of older systems need to type `search lan` twice because of a delay built into the boot servers.)

Your instructor will indicate which system you should boot from.

4. When (and if) prompted, Choose the appropriate keyboard. (Note: This will only happen if you're installing on a S700.)
5. Choose *Install HP-UX* from the menu that appears.
6. Enable networking, using the default routing IP address. The subnet mask here is 255.255.248.0.

-
7. Choose the Standard LVM configuration. *Do not* use the whole-disk (non-LVM) configuration. (Note: LVM is the default on S800, but non-LVM is the default on S700)

If the swap size is less than the amount of physical memory, make swap size larger so that it is at least equal to physical memory.

Make sure the remaining file system space is *not* allocated to /home. You will need the extra space for the LVM module.

Press *OK*. Now explore the cold-install screens, being careful to not change anything accidentally:

- a) How big is the /var volume by default?
- b) On the root file system, what is the free space threshold percentage set to?

Press *Cancel* to get back to the System Configuration screen. Press *OK* to accept the configuration.

8. *Do not* interact with *swinstall* for this exercise. Start the software installation from the SD server that your instructor indicates, using the directory /Depot there. While it runs, read ahead to the next question.
9. At this point, it will take about an hour to install the system software. Let your instructor know when you have gotten to this point.
10. The instructor will start the lecture for the next module. While you're listening, the software will load, the kernel will be rebuilt, and the system will come

up. You will see the “Welcome to HP-UX” `set_parms` screen with its questions.

Wait until the lecture ends (please do not type during the lecture!). Answer the questions of `set_parms`:

Yes, link to a network.

Set the time zone appropriately, and adjust the system clock time if necessary.

Set the root password to `root`.

Confirm that the IP address, subnet mask, and gateway address (it should be your own IP) are set properly.

Set up DNS, if necessary. (You should have written this information, if appropriate, down earlier.)

Do not set up NIS.

Skip the font client/server step. Make note of the amount of unused disk space below:

11. You can now log in as `root`. Manually — with `vipw`, not `SAM` — create a normal user account for you and your lab partner to share. `SAM` will be covered later. (Log in as the super-user only when you must.) What is the value of the `$EDITOR` environment variable, by default?

12. Simulate a customer recovering the user files.

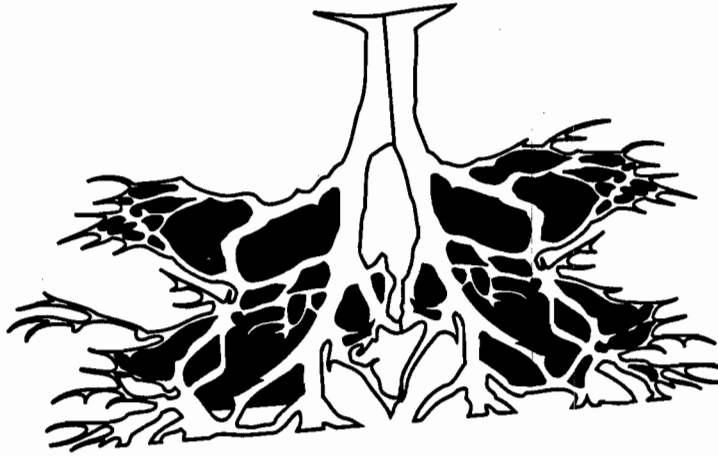
Run `ftp` to IP address 15.4.25.196, log in as `root` (same password as your system's root password), and get `/opt/deltaclass/opt.tar.Z`. Uncompress

and untar it on your system. To your `/etc/hosts` file, add the contents of

`/opt/deltaclass/lib/hosts.here.`

Be careful to eliminate duplicate host entries like your own machine's. Then remove the `opt.tar.Z` file.

File System Layout



Module goals:

The purposes of this module are to describe new locations for files, and to explain how to locate files under the new file system layout.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe motivation, features and benefits of changing the file system layout in HP-UX 10.0.
2. Complete a diagram or table illustrating the standard components of the HP-UX 10.0 File System Layout.
3. Describe transition links, and use commands to install and remove them.
4. List the new locations for given files.
5. Use new and classic tools to resolve pathname issues.

Motivation to change file system

- Conforms to SVR4 standards
 - *De facto* industry standard adopted by other major vendors
 - Supports HP's commitment to openness



Notes

- This file system has many names in industry.
 - AT&T SVR4 File System
 - V.4 File System
 - OSF/1 File System
 - V4FS
- In this module, it will be referred to as "HP-UX 10.0 File System Layout".
- Every major vendor, including HP, has slightly different implementations of this standard file system.
 - While the HP implementation is not "pure," it is compatible with the standard file system.
- Customers with inter-vendor networks will find the HP-UX 10.0 File System Layout is similar to that used by other vendors.
- When supporting multiple platforms, developers will require fewer HP-UX specific designs.
- *History*
 - Proprietary conventions

Features and benefits

Features

- Separates and groups files by functionality, *e.g.*,
 - Static vs. dynamic
 - Executables vs. configuration data
 - Operating system vs. applications
- Separates applications from each other

Benefits

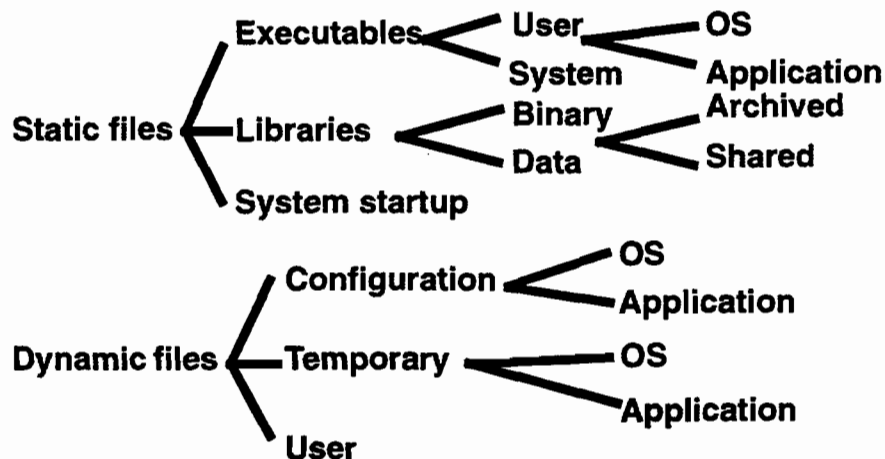
- Provides solid foundation for diskless and client/server file sharing models
- Simplifies system administration



Notes

- The HP-UX 10.0 File System Layout facilitates networked file sharing.
 - Files that are shared among hosts are separated from host-specific files.
 - Operating system files are separated from application files.
 - Applications are kept separate from each other.
- These groupings also benefit the system administrator.
 - Configuration files are kept separate from log files and temporary files.
 - Files that “grow” (dynamic) are restricted to a manageable subset of directories.
- How does this help the system administrator’s tasks?
 - System backups are easier.
 - Configuration changes are not lost when updating.
 - Disk space administration is simplified.

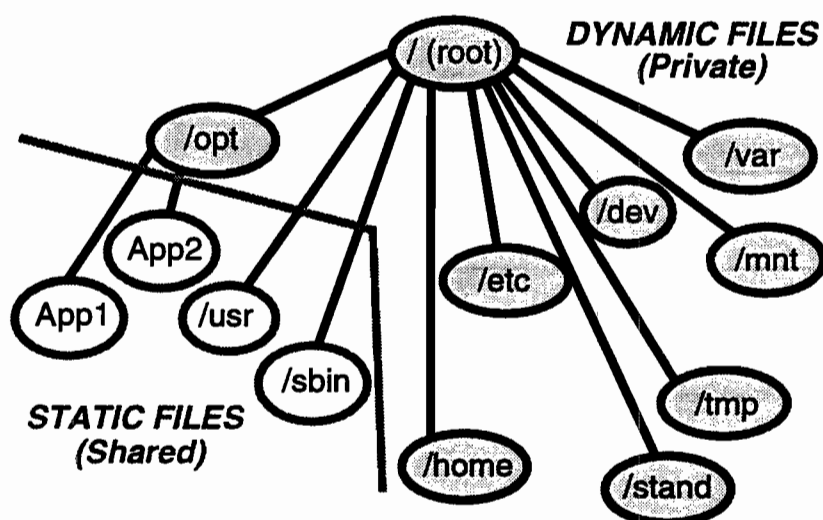
File system layout overview



Notes

- The dynamic portion of file system can grow and is separate from static areas.
- Static files are sharable by other systems.
- Dynamic files are also private or host-specific.
- This model separates host-specific configuration data from executable code using data.
 - This is integral to enable new diskless model.
- System administrators can backup, maintain, delete and alter files within their groupings, without affecting other file groupings.
- Prior to 10.0:
 - Static and dynamic in same directories (e.g., /usr/adm/)
 - Applications and OS files mixed in /usr (e.g., /usr/bin/, /usr/lib/uucp/)

Directory tree



Notes

- The directory tree above is described in more detail by the following table.

Directory name(s)	General description of contents
/usr, /sbin	Shareable portion of OS
/opt	Application subdirectories
/usr, /sbin, /opt/app1	Static, shareable among networked hosts—NO host-specific info.
All private/dynamic directories	Host-specific files
/etc	Host-specific configuration data essential to system operation. This directory no longer contains any executables.
/var	Host-specific files created during system execution
/home	Root for all users' home directories
/export	Root for sharable, networked file systems

Major new directories in 10.x

Name	Contents
/etc/opt/<application>	Application-specific configuration files
/etc/rc.config.d	Startup configuration files
/export	Default root of exported file systems
/opt	Root for optional applications
/sbin	Essential system commands
/stand	Stand-alone machine-dependent binaries and kernel configurations



Notes

- Each application's configuration files are stored in a separate subdirectory located under /etc/opt.
- All system startup configuration files are located in /etc/rc.config.d.
 - Details on this will be covered in the *System Startup* module.
- The sharable, static directory /sbin contains
 - commands necessary to boot and shutdown the system,
 - the file system mounting commands,
 - the very basic HP-UX commands, such as mv, cat, mkdir and sh.
- The private directory /stand contains system-specific kernel configuration and binary files.
 - These files, such as a generic kernel, are typically needed upon bootup.

Major new directories in 10.x

Name	Contents
<code>/usr/ccs</code>	Bundled development package
<code>/usr/lbin</code>	Backends to other commands
<code>/usr/old</code>	Obsolete files (was in 9.x, but used differently)
<code>/usr/sbin</code>	System administration commands
<code>/usr/share</code>	Architecture-independent sharable files
<code>/var</code>	Directory hierarchy for files created at runtime, e.g. log files and temporary files (variable size files)



Notes

- The `/usr/ccs` directory contains the minimal C compiler, with functionality sufficient to build a kernel, and related commands.
 - A subdirectory within `/opt` contains the fully-functional C compiler.
- The `/usr/lbin` directory stores backends to `/usr` hierarchy commands.
 - Commands such as lp filters and new trusted system utilities
 - Some networking daemons, such as remote server daemons and ftpd
 - Some subdirectories for special systems, such as `spell` and `uucp`
- The `/usr/old` directory is used for host customization, and contains a directory hierarchy similar to `/`. It includes
 - system files replaced by files in `/usr/newconfig`,
 - old versions of software for compatibility with a previous release.
- The `/usr/sbin` directory contains system-administration commands, many of which were in `/etc`, and documented in section 1M of the HP-UX Reference Manual.
- The `/usr/share` directory hierarchy contains files that can be shared among various architectures (e.g., terminfo files, kermi files, `spell` dictionaries, man pages, other HP-UX system documentation).

New names for old directories

10.0 Name	9.x Name
/home	/users
/opt/<application>	/usr/<application>
/usr/bin	/usr/bin + /bin
/usr/conf	/etc/conf
/usr/lib	/usr/lib + /lib
/usr/newconfig	/etc/newconfig
/usr/share/dict	/usr/lib/spell
/usr/share/man	/usr/man
/var/adm	/usr/adm
/var/adm/acct	/usr/adm/acct

Notes

- All user directories are created and managed under the dynamic /home.
- The /opt subdirectories each have a different application release in them.
- The subdirectories under /opt are shareable; /opt is not.
 - This allows each host to have its own set of applications
 - This also allows different releases of an application to be mounted separately.
- The /usr/bin directory contains common utilities and applications, primarily those documented in section 1 of the HP-UX Reference Manual.
- The /usr/conf directory contains the sharable kernel build environment.
- The /usr/newconfig directory has default system configuration data files. It contains a directory hierarchy similar to /.
 - The /etc/newconfig files may be in /usr/newconfig or in /opt/<application>/newconfig

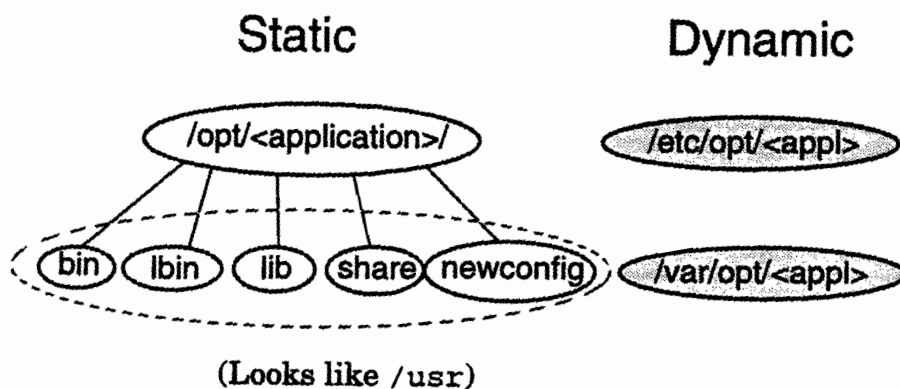
New names for old directories

10.0 Name	9.x Name
<code>/var/adm/crash</code>	<code>/tmp</code>
<code>/var/adm/cron</code>	<code>/usr/lib/cron</code>
<code>/var/adm/lp</code>	<code>/usr/spool/lp's spooler log files</code>
<code>/var/mail</code>	<code>/usr/mail</code>
<code>/var/news</code>	<code>/usr/news</code>
<code>/var/preserve</code>	<code>/usr/preserve</code>
<code>/var/spool/<...></code>	<code>/usr/spool/<...></code> (except for <code>/usr/spool/uucp</code> , see below)
<code>/var/tmp</code>	<code>/usr/tmp</code>
<code>/var/uucp</code>	<code>/usr/spool/uucp</code>

Notes

- The directory `/var/adm/crash` stores any kernel crash dumps.
- The `/var/adm/cron` directory contains log files maintained by `cron` and `cron FIFOs`.
- The `/var/tmp` directory holds user temporary files created by `/usr` hierarchy commands.
 - Temporary files created by `/opt/<application>` applications are in `/var/opt/<application>`.

Application directories



Notes

- Each application has its subdirectory under `/opt`, `/etc/opt`, and `/var/opt`.
- Applications are self-contained in their own sharable `/opt/<application>` directory, which has the same hierarchy as the operating system layout. For example:
 - The application's user commands are in `/opt/<application>/bin`.
 - The application's man pages are in `/opt/<application>/share/man`.
 - The application's libraries are in `/opt/<application>/lib`.
- The application's host-specific log files, and other temporary files, are stored under `/var/opt/<application>`.
- The application's host-specific configuration files are stored in `/etc/opt/<application>`.

Path variables

- System default-login files source special path-variable files for the system-path variables:

Filename	Variable name	Locations of
/etc/PATH	PATH	commands
/etc/MANPATH	MANPATH	man pages
/etc/SHLIB_PATH	SHLIB_PATH	shared libraries

- Software Distributor (SD-Ux) modifies these files upon software installation.

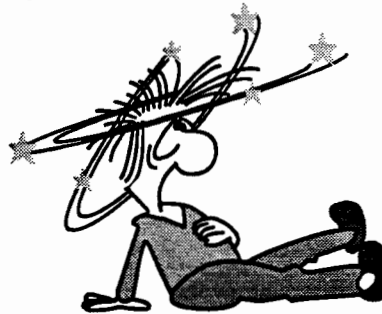


Notes

- Product binaries, man pages and other files are no longer contained in common system directories, such as /usr/bin and /usr/man.
- Default system path values do not point to all installed software and must be modified as products are added.
- Three new path variable files are provided to contain path variable values, as listed in the previous table.
 - The HP-UX system default login files, below /etc, source these files.
 - No applications, nor their installation scripts, should directly modify these files.
 - SD-UX provides configuration utilities for applications to use to add values to these paths.
 - **If an application does not use this automated mechanism, then users must ensure path variables are set correctly for their environment.**
 - SD-UX will be covered in more detail later in this course.
- Note that these files are read at login time by the shells.

Overview of impacts

- Everyone using HP-UX will be affected.
- Most files are re-located.
- Path variables and pathnames must be modified.
 - Applications
 - Source code
 - System administration files
 - User files
 - Documentation
- “Transition links” can help
- Several tools are available to find new paths and resolve path changes.



Notes

- Changes to the filesystem layout will affect every type of HP-UX user, as nearly every file has been moved to a new pathname.
 - User
 - System administrator
 - Developer
- HP-UX 10.0 provides tools to use prior to an upgrade from 9.0x to 10.0 that will help deal with file system conversion issues. These tools can:
 - investigate an existing 9.0x file system for potential problems with converting to the HP-UX 10.0 File System Layout
 - convert pathnames in shell scripts, makefiles and ordinary ASCII files to comply with HP-UX 10.0 File System Layout
- These pre-upgrade analysis tools were covered in the module “Pre-Upgrade Analysis Tools”.
- At HP-UX 10.0 there are “transition links” available.
- Additionally, HP-UX 10.0 provides tools to help locate files.
 - These tools will be covered in this module.

Transition links

- By default, installed on all HP-UX 10.0 systems
- Provide special-purpose symbolic link to map 9.0x file and directory names to 10.0 names
- Examples:

/bin	—————>	/usr/bin
/lib	—————>	/usr/lib
/bin/make	—————>	/usr/ccs/bin/make
/etc/checklist	—————>	/etc/fstab

- Differentiated from usual symbolic link by having its sticky bit set
- Example:

```
lr-xr-xr-t 1 root sys ... /etc/tar -> /usr/bin/tar
```

Notes

- Note that setting the sticky bit on the symbolic link doesn't affect its behavior, it just allows you to detect that this is a special type of link.

Transition link utilities

- The `tlinstall` and `tlremove` utilities create and remove transition links
- Not all 9.0x files and directories are mapped
- Transition links can be installed or removed on a fileset or group of fileset basis
 - As previously noted, by default, all filesets' transition links are installed on HP-UX 10.0
- Additionally, `tlinstall` will remove "dangling" transition links, which are 9.0x links specified in TTDB that point to non-existent locations

Notes

- The `tlinstall` utility will not create transition links for 9.0x file or directory names that have existing symbolic links.
- It will remove "dangling" transition links automatically and replace them with the correct transition link.
- If a transition link exists to a 10.0 location not specified in the TTDB, it will be converted to a standard symbolic link.
- For more information, refer to the man page for these commands.

Tools to locate files

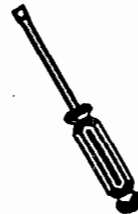
New tool

- `fnlookup`



Classic tools

- `find`
- `which`
- `whereis`
- `whence`
- `strings`
- `man`
- `ls`



Notes

- Each of these tools will be covered in more detail on the next pages.
 - Please also refer to each tool's man page, or to the HP-UX documentation, for additional information.

Using fnlookup to locate files

- “Filename lookup utility”
- Temporary 9.0x to 10.0 update aid
- Provides 10.0 pathname for 9.0x pathname
- Provides 9.0x pathname for 10.0 pathname
- Syntax example:

```
fnlookup -o <9.0x pathname>
```

```
fnlookup -n <10.0 pathname>
```



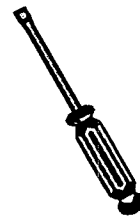
Notes

- The filename lookup utility provides file system pathname mappings between the HP-UX 9.0x and HP-UX 10.0 directory structures.
- Its data is maintained in a data file called the Transition Tools Database (TTDB).

Filename	9.0x default location	10.0 default location
fnlookup	/upgrade/bin	/opt/upgrade/bin
TTDB	/upgrade/lib	/opt/upgrade/lib

Using classic tools—a quick review

```
find      find / -name <filename> -print
which     which <filename>
whereis   whereis <filename>
whence    whence [-v] <filename>
strings   strings -a <filename>|grep /| sort -u
man        man <filename>
ls         ls
```



Notes

- These tools were available on 9.0x. This page should be review.
- Each syntax example above shows how to use the command to locate a file. Many of these commands provide additional information about the file.
 - Please refer to each command's man page for additional options and a detailed description.
- Use `find` to locate any file on the system.
- Use `which` to locate any executable file located in the user's PATH.
- Use `whereis` to locate any file located in system `src`, `bin`, `man`, `lib`, etc, include and games directories.
- Use `whence` , supported in the POSIX or Korn shell only, to locate any executable file located in the user's PATH.
- Use `strings` to look for any pathnames that are referenced in a file, whether it is a binary, text or data file.
- Use `man` to locate information about any file for which there is a man page.
- Use `ls` to list files in the current directory.

Summary and review

- The HP-UX 10.0 File System Layout conforms to SVR4 and OSF/1 models
- *For NFS Diskless support (post-10.0):*
 - *Supports diskless and client / server file sharing*
 - *Separates sharable from host-specific files*
- Simplifies system administration
 - Separates configuration files from temporary and log files
 - Separates dynamic files from static files
- Nearly every file is located in a new path
- Nearly every user is affected
- Transition links and tools help locate files

Notes

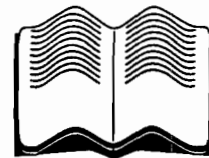
- The HP-UX 10.0 File System Layout makes it easier to:
 - transition from other standard systems
 - administrate multi-vendor systems
 - administrate applications
 - backup files
 - administrate disk space

For more information



For more information

- *HP-UX 10.0 File System Layout Whitepaper Version 2.2, 9/29/94*
- The man pages for the commands described
- The hier(5) man page



Notes

- This whitepaper is located in this module's "Additional Notes".

A customer calls you after she installs HP-UX 10.0. She used to have 9.0x, but decided to use 10.0 on some new systems. She tells you she can no longer find any of the files where she expects them. Help her out by answering the following questions:

1. Why was the HP-UX File System Layout changed?
(summarize standards/features/benefits)

2. What are the major separations of files?
Complete the following table (to fax to her).
(Remember: two names for each side of the division)

	Static	vs.	Dynamic
Definition			
Examples of files stored here			

	Shared	vs.	Private
Definition			
Examples of files stored here			

3. Where is everything now?

Complete the following table (to fax to her), to show _____
where some of the major files are located now.

Contents	Directory (or filename)	Example of contents (filename)
Kernel		
Required system commands		
System admin. commands		
Configuration data required for operating system		
man pages for OS		
most spooling directories		
Variable size files created during runtime		
Applications		
man pages for application		
Users' home directories		

4. What are the names of and purpose of the system
path variables?

What is the alternative to using these?

Show the customer how to locate new files. Demonstrate by using `fnlookup` to answer the following questions: _____

1. What was the 9.0x location of `cat`? _____
Where is it located now? _____
List three other common commands located here:

2. What was the 9.0x location of `cron`? _____
Where is it located now? _____
What was the 9.0x location of `lpsched`? _____
Where is it located now? _____
List three other common commands located here:

3. What was the 9.0x location of `init`? _____
Where is it located now? _____
4. What subdirectories are in `/usr/share`?

5. List three files in `/stand`.

6. List three directories in `/var`.

7. View the directories in `/opt`.
Which have matching directories in `/etc/opt`?

Which have matching directories in `/var/opt`?

8. Now is your time to explore! Find commands or utilities that you use most often. Where are they at 10.0?

Additional notes for "File System Layout" module

This page intentionally left blank.

User Environment



Module goals:

The purpose of this module is to ensure that the student understands and is able to describe changes in areas that directly affect users.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. List major differences between POSIX shell and Korn shell.
2. Run user-level commands that have changed in this release.
3. Describe changes to audio.
4. Use the new terminal emulator.

Motivations for user-level changes

- Standards supported by HP-UX 10.0 affect many areas of the user environment, including:
 - various user-level commands
 - the default shell
- Customers expect easy access to multimedia (e.g., audio)



Notes

- Standards affecting user environment include
 - OSF/1
 - POSIX
 - XPG4
 - SVID3



Features and benefits

Features

- More powerful default shell—the POSIX shell
- Additional and enhanced user-level commands
- Audio control on VUE Front Panel
- New terminal emulator dtterm

Benefits

- Default user shell has more capabilities
- Users have access to more standard commands
- User can manipulate audio features
- User can easily modify dtterm terminal emulator window's display



Notes

Default shell is POSIX shell

- Superset of Bourne shell
- Enhancement of Korn shell
- Replaces Bourne and Korn shells

Description	Pathname
POSIX shell	/usr/bin/sh
Korn shell	/usr/bin/ksh
Bourne shell	/usr/old/bin/sh

- 4 byte EUC support
- XPG4 conformance



Notes

- Use of the POSIX shell assures portability across POSIX-compliant operating systems.

POSIX shell features not in ksh

- `hash -r` removes all previously remembered external command locations
- `export -p` writes names and values of exported variables to stdout
- `umask -S` prints current mask value in symbolic format
- `unalias -a` removes ALL alias definitions from shell environment
- `unset -v varname` unassigns parameters of *varname*
- `/usr/bin/sh -c` spawns a process for this
- `command` executes commands without doing the function lookup
- Does function lookup before built-in commands

Notes

- The hash command is now built-in shell command, with new `-r` option
- For the command `export -p` use proper quoting in format, so is suitable for re-input to shell as commands that achieve same exporting results.
- Some additional changes include:
 - The type command is now a built-in shell command.
 - The time keyword is unavailable, so use the POSIX compliant `/usr/bin/time` utility.
- Refer to shell documentation for list of differences between ksh and POSIX shell, for revising shell scripts.

Modified user-level commands

Command	Remarks
cat(1)	New options -b, -n, -r; options -e, -t no longer need -v option
chown(1) chgrp(1)	Default behavior on both S700 and S800 is to follow symbolic link; use -h to <u>not</u> follow symbolic link
cksum(1)	Uses new algorithm, so will output different value than on 9.0x for same input; allows piped input
compress(1)	new option -z is same as -f, except does not force compression for null compression

Notes

- The cat command has three new options:
 - b Omits line numbers from blank lines.
 - n Displays lines preceded by line numbering.
 - r Replaces multiple consecutive empty lines with 1 empty line.

Modified user-level commands

Command	Remarks
<code>sh-posix(1)</code>	Enhanced version of <code>ksh</code> ; default shell; Bourne shell replaced by this
<code>sort(1)</code>	New option <code>-A</code> to sort on a byte-by-byte basis, removed <code>-l</code> option, enhanced <code>-f</code> and <code>-t</code> options to support multibyte chars.
<code>sum -p(1)</code>	refer to <code>cksum(1)</code>

Notes

- The `sed` text editor outputs usage message if bad options are used.
 - The output of its `"l"` command does not split characters across lines.
- The `sort` command's new option `-A` sorts on a byte-by-byte basis using each character's encoded value.

Modified user-level commands

Command	Remarks
<code>tput(1)</code>	SVR4: Supports <code>-S</code> option to read from <code>stdin</code> , and 59 terminal capabilities added
<code>tr(1)</code>	XPG4: Berkeley-style, not Bell command line--new syntax, old usage syntax is obsolete; new <code>-A</code> option to translate on byte-by-byte basis; many changes to abbreviations used to introduce characters
<code>vi(1)</code>	refer to <code>ex(1)</code>

- See man pages and *HP-UX Reference* manual for additional information about any commands.

Notes

- The `tr` command has new usage syntax:
 `tr [-Acs] string1 string2`
 `tr -s [-Ac] string1`
 `tr -d [-Ac] string1`
 `tr -ds [-Ac] string1 string2`
 - Using `-A` option with `tr` means it will not support extended characters.
- Other section(1) commands that have changed or are new are listed in this module's "Additional Notes" section.

Modified user-level commands

Command	Remarks
<code>doschmod(1)</code>	All DOS commands accept wildcards in
<code>doscp(1)</code>	DOS filenames:
<code>dosls(1)</code>	Metacharacters "*", "?", and "[. . .]"
<code>dosrm(1)</code>	can be used when specifying DOS file names,
<code>dosrmdir(1)</code>	using the DOS commands.
<code>echo(1)</code>	Recognizes two more special characters (<code>\a</code> , <code>\0num</code>)
<code>ex(1)</code>	:tag has new tag stacking feature; :pop is new
<code>lp(1)</code>	support for PCL5 features for new models

Notes

- The `echo` command recognizes two additional special characters.
 - `\a` = write an alert character
 - `\0num` = write 8-bit value that is zero-, one-, two- or three-digit octal number *num*
- The new `lp` models are listed in this module's "Additional Notes" section.

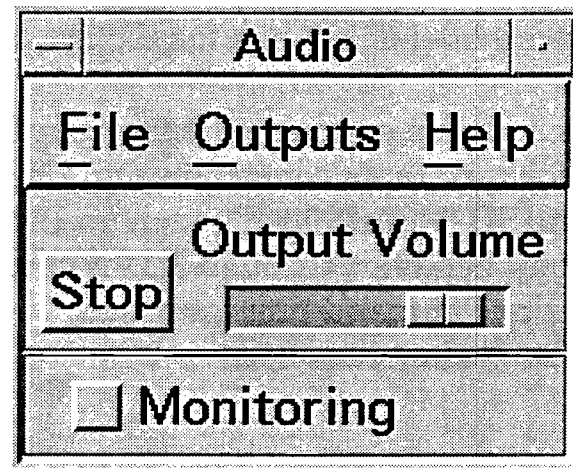
Modified user-level commands

Command	Remarks
<code>more(1)</code> <code>page(1)</code>	XPG4: New syntax, new options and interactive commands, new default behavior (e.g., does not stop at ^L); can scroll backwards; can examine another file, and many other new features and capabilities
<code>rm(1)</code>	can descend to arbitrary depths in file hierarchy and will not fail due to path length limitations; -f, -i, -r and -R options change behavior
<code>rmdir(1)</code>	-f and -i options change behavior as with <code>rm</code>

Notes

- The `more` command has new syntax, including many new options:
`more [-n] [-cdefisuvz] [-n number] [-p command] [-t tagstring]
[-x tabs] [-W option] [+linenumber] [+/pattern] [name ...]`
- For `rm` and `rmdir`, `-f` ignores any previous `-i` occurrence and vice versa.

VUE Audio Control Panel

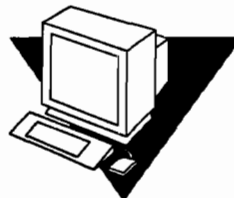


Notes

- VUE's Front Panel has new control in bottom row: the audio control.
 - This control only displays if the system has audio capability.
- Clicking the audio control causes the Audio Control panel to display.
 - Allows control over volume, source, and destination of external audio sources (e.g., compact disk player)
- Stereo is available.
- Test sounds are available in /opt/audio/sounds.
- The Audio Editor has an improved GUI and some new menu options.
 - *File...New, File...Save, Options...Save Settings*
 - *Auto Gain Control* to automatically adjust Record Level slider during recording to avoid distortion
- Some menu items are no longer available:
 - *Output Device* replaced by Audio Control Panel *Outputs*
 - *Write Data Format* replaced by selections within *Save As...* dialog box

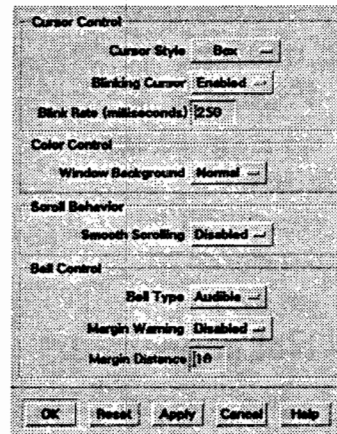
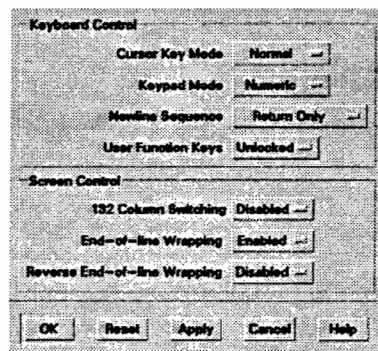
The dtterm terminal emulator

- New terminal emulator for HP-UX 10.0
- Must use instead of hpterm for EUC 4-byte codesets
- Provides runtime support of legacy applications written for terminals conforming to ANSI X3.64-1979 and ISO 6429:1992(E), such as DEC VT220
- Can specify behavior through X11 resources or through the command line interface
- Located in `/usr/vue/bin`



Notes

The dtterm window options



Notes

- The dtterm window has a menu which allows access to options to control various aspects of the terminal emulator.
 - Keyboard
 - Cursor
 - Bell
 - Scrollbar
 - Color
 - Font

Tips and tricks

- To access Bourne shell, add to path:
/usr/old/usr/bin
- To use tput -S:
 - Example 1 – Example 2
 - tput -S << EOF tput clear cup 10 20 bold
 - clear
 - cup 10 20
 - bold
 - EOF
 - Not supported:
 echo clear cup 10 20 bold|tput -S
- “Audio” in Style Manager controls bell,
not Audio Control Panel or speaker



Notes

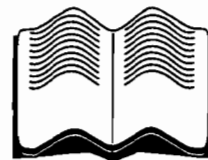
Summary and review

- Default shell = POSIX shell
- User-level commands modified to conform to various standards
- Audio Control Panel added to VUE Front Panel
- New terminal emulator dtterm provides easy access to display options

Notes

For more information

- The man pages for individual commands
- List of other modified and new commands from Section 1 located in “Additional Notes” section
- HP Visual User Environment 3.0 User’s Guide (for HP-UX 10.0)



Notes

These simple exercises guide you through using some new and modified user environment features. Experiment with other features discussed in this module, if you'd like.

1. From the POSIX shell, enter the following sequence of commands, and note the output of each. Write the output down for the indicated commands.

```
ls
```

```
hash
```

What is the output?

```
alias
```

```
hash -r
```

```
hash
```

What is the output?

```
alias
```

```
ls
```

```
hash
```

```
unalias -a
```

```
hash
```

What is the output?

```
alias
```

What is the output?

2. Enter `export -p`.

What is the HOME variable value? _____

What is the PATH variable value? _____

Use `unset -v` to unset the MANPATH variable.

Run `export -p` again to check it. _____

-
3. Run `/usr/bin/sh -c "ps -f"`. (`-c` is new option)

What is the PID of this process? _____

What is the name of and PID of its parent process?

4. What is the current mask value in symbolic format?

What command did you enter to do this?

5. Run `cat -n` on `/etc/services`.

How many lines? _____

6. Run `more -p G` on `/etc/services`.

What displays? _____

Enter `h` for help on `more`'s interactive commands,
many of which are new.

What do you enter to go backward one screen?

What does entering `=` do?

7. Click audio button and view the Audio Control Panel's
menu options. What can you do from here?

8. Run `dtterm`.
Disable blinking cursor and inverse the color.
Change the font size to a smaller size. Experiment!

Additional notes for “User Environment” module

This page intentionally left blank.

Other modified or new
commands from Section 1



The following commands from section(1) have also
changed. Refer to command's man page for information.

Changes to user-level commands:

paste	uniq
join	mm
mesg	sar

Changed to support, or enhance support, of international-
ization (e.g., NLS, multi-byte character support):

adjust	gencat
iconv	locale
nljust	nroff

Changes will affect developers, and sophisticated users:

awk	sed
make	sccshelp (was help)
pax	

Changes affecting superuser usage of command:

passwd

(changes to passwd are covered in "SAM and Administra-
tive Commands" module)

New:

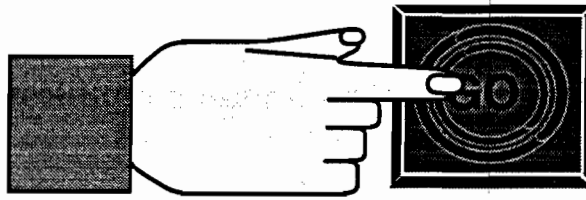
dmpxlt	genxlt
sccs	

New model files include support for these printers:

PaintjetXL300	Deskjet1200C
Laserjet4	Laserjet4Si
Deskjet500C	Deskjet550C
Laserjet4p	

This page intentionally left blank.

System Startup



Module goals:

The purpose of this module is to understand the system startup sequence well enough to modify it at the most appropriate point.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the features and benefits of the new startup/shutdown model.
2. List directories used in the system startup sequence, and the files in each.
3. Describe purpose, usage and contents of each type of script in this model.
4. Explain the relationship between each of these scripts.
5. Interpret scripts' names and explain their naming conventions.
6. Describe how and when startup/shutdown scripts are invoked.
7. Define run levels and explain their relationship to startup/shutdown.
8. Modify system startup/shutdown operations.
9. Add a module to the system startup/shutdown sequences.

Motivations for changing model

- **Follow industry-standard mechanism**
 - Sun, SGI, AT&T all use similar scheme
 - Adopted from OSF/1
- **Separate startup (rc) scripts into functional areas**
 - Eases maintenance through simplification of scripts
 - Easier to troubleshoot
- **Break system administrators of habit of modifying scripts for better supportability**

Notes

- **This startup/shutdown paradigm is a standards-based solution (OSF/1) to control subsystem initialization and termination.**
- **It enables a consistent and standard method for developers, system administrators and users to add, modify, and/or delete startup/shutdown behavior.**

Features and benefits of new model

Features

- Execution scripts separated from configuration
- Behavior of startup/shutdown sequence modified by changing configuration variables
- Each subsystem has its own execution script
- Each subsystem can be started/stopped on a per-init-level basis

Benefits

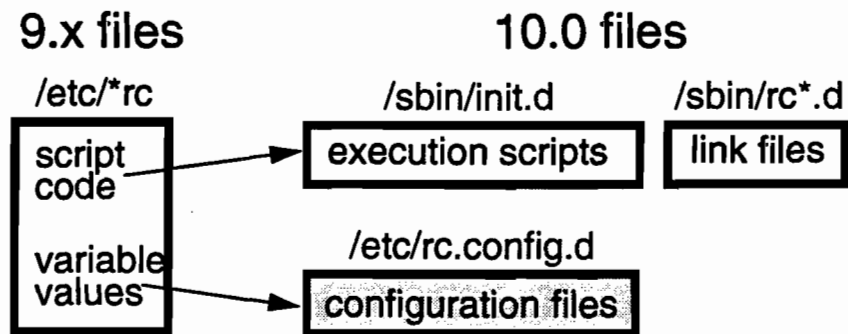
- Provides easier system administration and troubleshooting
- Enables fine levels of control and subsystem separation



Notes

- Execution scripts are located in `/sbin/init.d`.
- Separate configuration variable scripts for behavior modification are located in `/etc/rc.config.d`.
- Execution scripts should not be modified.
- Instead, configuration variables in the configuration variable scripts are changed to modify system startup/shutdown sequence activity.
- There is a one-to-one mapping between scripts and functional subsystems.
- Directories of scripts control startup/shutdown sequence.
 - Separate directory for each run level, controlled by link files.
- The execution scripts can now be replaced, instead of merged, at each system update, which should make upgrading easier.

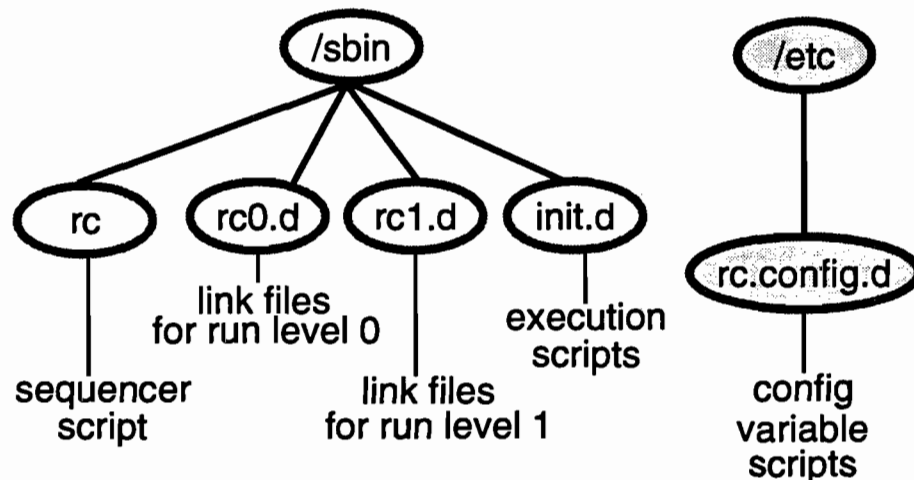
9.0x vs. 10.0 scripts



Notes

- The 9.0x /etc/*rc scripts are obsolete and can be written over at 10.0.
- Modifiable scripts do not exist at 10.0.
- The /etc dynamic directory is for configuration data only.
- The /sbin static directory is where essential system commands exist.
- There are three components to the startup/shutdown configuration in 10.0:
 - Execution scripts
 - Link files
 - Configuration variable scripts

Startup/shutdown model overview



Notes

- **Configuration variable scripts**, located in `/etc/rc.config.d`, are modified to set variables that affect whether a subsystem is enabled, disabled, or has other modifications made to it at startup/shutdown.
- Each configuration variable script has a matching execution script.
 - Usually it has the same name as the execution script.
- **Execution scripts**, located in `/sbin/init.d`, read variables from configuration variable scripts and run startup/shutdown commands.
- There is one execution script for every functional subsystem.
 - Usually it has the same name as the subsystem.
- **Link files**, located in `/sbin/rc*.d`, are used to control the sequencing order of execution scripts.
 - Link files are symlinks to execution scripts to be executed on transition to a specific run level.
- Each run-level has its own link file directory.
- The **startup/shutdown sequencer script**, `/sbin/rc`, invokes execution scripts based on run-level transitioning and the link file sequence.

Example execution script: cron

```
case $1 in
start_msg)
    echo "Start clock daemon"
    ;;
stop_msg)
    echo "Stop clock daemon"
    ;;
'start')
    if [ -f /etc/rc.config.d/cron ] ; then
        . /etc/rc.config.d/cron (...)
    'stop')
        pid=`ps -ef | awk 'NF~/cron/ {print $2}'`
        if [ "X$pid" != "X" ]; then
            if kill $pid; then
                echo "cron stopped"
            . . .
```

Notes

- Execution scripts perform both startup and shutdown functions for the associated subsystem.
- How does an execution script know whether to startup or to shutdown the subsystem?
- The startup and shutdown sequencer script, /sbin/rc, invokes each execution script with one of four arguments, indicating the "mode":
 - start Bring subsystem up (start)
 - start_msg Report what start action will do (displays message)
 - stop Bring subsystem down (kill)
 - stop_msg Report what stop action will do (displays message)
- The start and stop messages display on the console during startup and shutdown, as the system startup/shutdown checklist.

System startup checklist

- Partial view of the checklist:

HP-UX Startup in progress

```
Mount file systems ..... [ OK ]
Setting hostname ..... [ OK ]
Set privilege group ..... [ OK ]
Display date..... [ OK ]
Save system core image if needed .....[FAIL]*
Enable auxiliary swap space ..... [ OK ]
Start syncer daemon ..... [ OK ]
Configure LAN interfaces ..... [ OK ]
Start Software Distributor agent daemon. [ OK ]
```

Notes

- The system startup and shutdown checklists obtain their display messages from the execution scripts' responses to the start_msg and stop_msg arguments.
- There are three values possible:
 - OK The execution script started up (shutdown) the subsystem.
 - FAIL A problem occurred with starting up (shutting down) the subsystem.
The asterisk after this value refers to a note at the bottom of the console screen:
 - * - An error has occurred !
 - * - Refer to the file /etc/rc.log for more information.
 - N/A The system administrator did not configure the script for the subsystem to start.

Running scripts: /sbin/rc

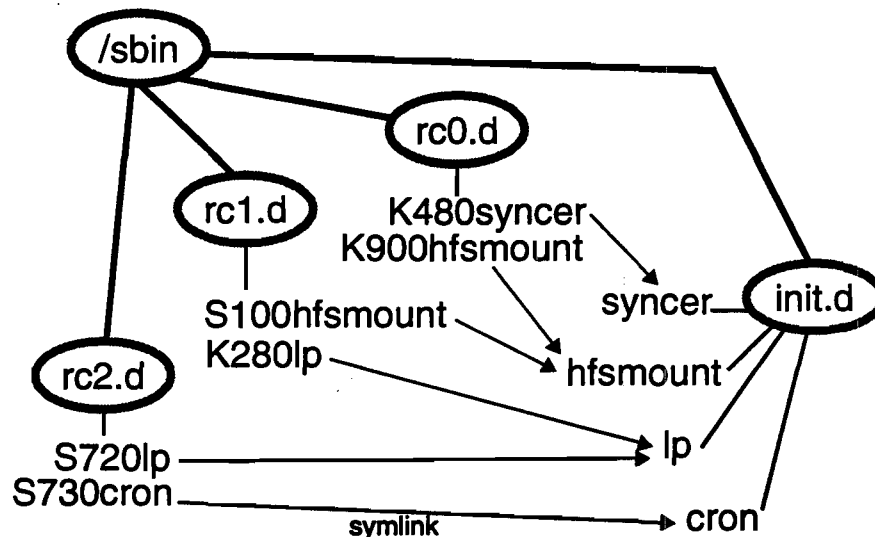
- The startup and shutdown sequencer script, /sbin/rc, runs execution scripts.
- Invoked by init when transitioning to run levels
- Runs execution scripts as indicated by /sbin/rc*.d link files
 - These determine which execution scripts are run, and in what order
- Runs execution scripts in “start” or “kill” mode, depending on direction of run level transition



Notes

- The script /sbin/rc runs the /sbin/init.d execution scripts, based on the contents of the /sbin/rc*.d directories.
- If transitioning from lower to higher run level, /sbin/rc runs execution scripts in “start” mode.
- If transitioning from higher to lower run level, /sbin/rc runs execution scripts in “kill” mode.
- Whether or not a subsystem is started depends on configuration variables in configuration variable scripts.

The /sbin/rc*.d directories



Notes

- These directories, also referred to as the “run-level directories” or “sequencer directories”, contain only symbolic links (“link files”) to the /sbin/init.d execution scripts.
- Actually, /sbin/rc “runs” these symbolic links to run the execution scripts when transitioning to run levels.
- Each run level has its own /sbin/rc*.d directory.
 - The number in * position indicates run level.
 - For example, /sbin/rc0.d contains link files to execution scripts that are run when run level 0 is entered.
 - Level 0 and level S are considered the same. The /sbin/rc0.d directory services both run levels.
- Because /sbin/init.d execution scripts must startup and shutdown their associated subsystems, they will have two links in the sequencer directories, a start link and a kill link.

Link files—"S"tart and "K"ill links

- Symbolic links to `/sbin/init.d` execution scripts

Start link

- Name starts with "S"
- Invoked with `start` argument at system boot or on transition to higher run level

Kill link

- Name starts with "K"
- Invoked with `stop` argument at system shutdown, or on transition to lower run level



Notes

- If an execution script has a start link in a sequencer directory, it is referred to as a "start script". If an execution script has a kill link in a sequencer directory, it is referred to as a "kill script".
- The type of link indicates whether it is used to run execution script in "start" mode or "kill" mode.
 - Start mode indicates that `/sbin/rc` passes `start` to the script
 - Kill mode indicates that `/sbin/rc` passes `stop` to the script

Start and kill link naming

- Symbolic links follow a strict naming convention
- Example:

/sbin/rc2.d/S060cron

1

2

3

4
- Each naming component has different meaning:
 - 1** = Run level number of sequencer directory
 - 2** = Sequencing type of sequencer link name
 - S = start link
 - K = kill link
 - 3** = Sequencing number
 - Used to sequence link files in sequencer directory
 - 4** = Script name
 - Same name as execution script to which it is linked

Notes

- Link files are executed in lexicographical order, using entire link file name.
- All link files in a sequencer directory are numbered to ensure a particular execution sequence.
 - Each link has, as part of its name, a three digit sequence number.
 - Sequence numbers identify a particular run-level task or service.
 - For a listing of sequence numbers and their meaning, refer to “HP-UX 10.0 File System Layout Whitepaper”, in the “File System Layout” module’s “Additional Notes” section.
- Start and kill link’s sequence numbers allow gaps so that adding a new script (symbolic link) does not mean re-numbering existing scripts.
- The name of the /sbin/init.d execution script to which it is linked completes the symbolic link name.
- Start and kill links that come from HP should **not** be re-named or moved:
 - Newly installed releases will overwrite scripts, anyway.
- Files, including symbolic links, from HP should **not** be erased or modified:
 - Use configuration variables to control script execution.

Script names and locations

- Subsystems are killed in opposite order as started, so kill scripts do not have same sequence numbers as start script counterparts.
 - Ordering of kill scripts is reverse of their start script counterparts
 - For example, S111flub followed by S211uses_flub have kill counterparts K511uses_flub followed by K711flub
- Start scripts in /sbin/rcNd have kill script counterparts in /sbin/rc(N-1)d.
 - For example, /sbin/rc3.d/S123flub and /sbin/rc2.d/K654flub

Notes

- If two subsystems are started in a given order due to dependencies, the kill script counterparts are numbered so that the subsystems are stopped in the opposite order from which they were started.
- Start scripts have kill script counterparts in the sequencer directory one level lower.

Example config variable script

```
#!/sbin/sh
# @(#) $Revision: 72.3 $
# Savecore configuration
#
# SAVECORE:Set to 1 to save system core files
# in $SAVECORE_DIR
# SAVECORE_DIR:Directory name for system core
# files. NOTE: The filesystem in which this
# directory is located should have as much
# free space as your system has RAM. This
# directory will be created by the
# /sbin/init.d/savecore script if it does not
# exist.
SAVECORE=0
SAVECORE_DIR=/var/adm/crash
```

Notes

- All configuration data affecting control of subsystems at system initialization and termination is stored in configuration variable scripts located in /etc/rc.config.d.
- Information in these scripts indicates whether to enable or disable the subsystem, and/or how to modify the subsystem.

Modifying config variable scripts

- **Edit these instead of execution scripts**
 - Use any text editor
 - SAM may also modify the configuration variable scripts
- **In most scripts, the first variable is named the same as the subsystem, and indicates whether to enable or disable it**
 - 1 = Enabled
 - 0 = Disabled
- **Comments provide information about how and what to modify**

Notes

- **Instead of modifying execution scripts, system administrators modify the configuration variable scripts to affect startup and shutdown.**
- **System administrators can edit these scripts using any text editor.**
- **Most scripts provide comments in the file with information about variables and their appropriate values.**

Run level definitions

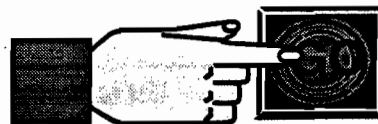
Run level	Definition	Sequencer Directory
0	Halted run level	/sbin/rc0.d
S	Single user state	/sbin/rc0.d
1	Boot-time system configuration	/sbin/rc1.d
2	Multi-user state	/sbin/rc2.d
3	Export filesystems	/sbin/rc3.d
4	VUE Startup	N/A (out of inittab)
5,6	Not currently used	N/A

Notes

- When run level 0 is entered, /sbin/rc executes **all** kill and start scripts in /sbin/rc0.d and then the system is halted.
 - These are quick system administration scripts that prepares the system for a shutdown.
- When run level S is entered, /sbin/rc executes **all** kill and start scripts in /sbin/rc0.d, leaving the system in single-user mode.
- When another run level is entered from a lower state, all start scripts in that run level directory are executed.
- When another run level is entered from a higher state, all kill scripts in that run level directory are executed.
- Run level 1 is used for setting necessary configuration values in /etc/rc.config.d/* if not already set and for performing host-specific configuration for newly installed software.
- Run level 2 is used for bringing the system to a multi-user state.
- Run level 3 is the default init state. It starts up the NFS server.
- Other run levels are configurable on a per-host basis.
- Note that Level 3 is the S800 default, and Level 4 the S700 default.

Example startup sequence

1. All start scripts are executed in `/sbin/rc1.d`.
 - Mount local file systems
 - Check/set critical system configuration values
 - Configure newly installed system software
2. All start scripts are executed in `/sbin/rc2.d`.
 - Start system daemons
 - Start networking
3. All start scripts are executed in `/sbin/rc3.d`.
 - Start NFS daemons to export/mount NFS



Notes

- When transitioning from run levels, `/sbin/rc` runs the appropriate start and kill scripts for the new run level, and all intermediate levels between the old and new.
- For each sequencer directory, or run level, (except level 0 and level S),
 - Start scripts are executed when entering directory from a lower level.
 - Kill scripts are executed when entering directory from a higher level.
- For example:
 - When booting up to run level 4, `/sbin/rc` runs the start scripts in `rc1.d`, `rc2.d`, `rc3.d` and `rc4.d`, in that order.
 - When shutting down from level 2, `/sbin/rc` runs the kill scripts in `rc2.d` and `rc1.d`, in that order, and then enters level 0.
- Entering level 0 or level S is the exception. In these cases, `/sbin/rc` runs **all** scripts in `rc0.d`.
- Additionally, level 0 or level S sequencer scripts are run **only** if the system is being shutdown or going into single user mode.

Adding startup/shutdown scripts

1. Create execution script for subsystem.
 - Follow template `/sbin/init.d/template`.
 - Name execution script same as subsystem.
 - Place execution script in `/sbin/init.d`.
2. Create configuration script for subsystem.
 - Refer to `/etc/rc.config.d` scripts as models.
 - Name same as execution script.
 - Place script in `/etc/rc.config.d`.
3. Create start and kill links.
 - Place in sequencer dir., based on dependencies.
 - For sequence numbers, refer to “HP-UX 10.0 File System Layout Whitepaper”

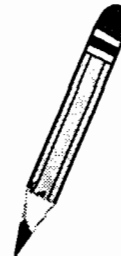


Notes

- Developers should write startup/shutdown scripts under this new model for their products requiring this.
- There is no “conversion” process for the 9.0x *rc scripts, rather developers should create new scripts as in the steps above.

Writing execution scripts

- See example: `/sbin/init.d/template`.
- Ensure script recognizes these four arguments:
 - `start_msg` Display message used by startup checklist
 - `stop_msg` Display message used by shutdown checklist
 - `start` Start subsystem
 - `stop` Stop subsystem
- Use `stdin/stdout` for all input and output.
 - Do not write directly to devices.
- Use **only** these standard exit values:
 - 0 Exited without error
 - 1 Encountered errors
 - 2 Skipped, didn't do anything
 - 3 Executed normally, must reboot for changes to take effect



Notes

- If passed `start_msg`, report action to be taken if `start` is passed.
 - No action other than displaying message should occur.
 - Message should be a single line with no more than 30 characters.
- If passed `stop_msg`, report action to be taken if `stop` is passed.
- Do not echo messages indicating entry or exit from script.
 - The `/sbin/rc` script will use `start_msg` and `stop_msg` and will record their messages.
- Direct status messages to `stdout` and error messages to `stderr`.
 - Both `stdout` and `stderr` are redirected to log file `/etc/rc.log`.
 - If startup checklist set to raw mode, `stdout` and `stderr` go to console.
- Do not use console during system boot or shutdown to display messages.
 - Console output during these times will overwrite graphical checklist.
- Use only the exit values indicated.
 - Returning any other value may cause script to appear to have been skipped in checklist, or may cause system to reboot.

Creating start and kill links

- Customers can use “Do Not Care” start link:
`/sbin/rc2.d/S900exec_file`
- Customers can use “Do Not Care” kill link:
`/sbin/rc1.d/K100exec_file`
- “Do Not Care” numbers mean order of startup/shutdown is not important for this process
- HP may also use these for “Do Not Care” links
- Developers should contact HP if they would like a “protected” sequence number assigned to them



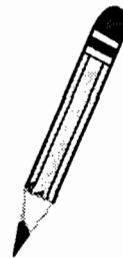
Notes

- Use these numbers if specific ordering of starting the subsystem is not a concern.
 - That is, if subsystem can be started at any point after system boot and initialization
- To create these links:

```
ln -s /sbin/init.d/exec_file /sbin/rc2.d/S900exec_file
ln -s /sbin/init.d/exec_file /sbin/rc1.d/K100exec_file
```
- For processes that require a particular startup/shutdown order, ISU developers should contact the below for a unique number:
PA-RISC Developers' Program
(508) 436-5144
pard@apollo.hp.com
 - Ask for information regarding “Startup/Shutdown Sequence Number Assignments”

Writing config variable scripts

- Use other `/etc/rc.config.d` scripts as model.
- Develop for POSIX shell.
- Do not refer to other scripts' variable definitions.
 - No sourcing order requirements supported.
- Keep each variable definition on separate line.
 - Other scripts or SAM may read or modify
- Place comments on separate line with #.
- Name the same name as the associated execution script.
- Be very careful, as errors in configuration scripts can cause boot problems



Notes

- Configuration variable scripts are created and managed by the developer of the subsystem.
 - Many configuration variable scripts come as part of HP-UX 10.0.
- Write scripts for default POSIX shell, not for the Bourne, Korn or C shells.
- Do not require any sourcing order for scripts, as they can be executed either individually, or all at once.
- Sourcing `/etc/rc.config` will source all the configuration variable files below `/etc/rc.config.d`.
 - The `/etc/rc.config` script will also source `/etc/TIMEZONE` which contains the TZ environment variable definition.
- Comment lines begin with a # symbol and are on separate lines from variable definition lines.
- No trailing comments are allowed on variable definition lines.

Summary and review

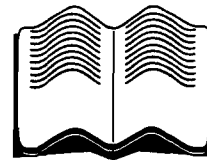
- New model replaces 9.0x /etc/*rc

Component	Purpose and comments	Location
Execution scripts	Perform subsystem startup and shutdown Are not customer-modifiable	/sbin/ init.d
Configuration variable scripts	Provide configuration variables to affect execution script behavior - sourced by execution scripts	/etc/rc. config.d
Start scripts	Symbolic links to execution scripts to be run in "start" mode at a particular run level Naming convention sets up execution sequence	/etc/rc*.d * = run level
Kill scripts	Same as start scripts but run in "kill" mode	same as above
/sbin/rc	Runs execution scripts based on rc* .d files	/sbin/rc

Notes

For more information

- *HP-UX 10.0 File System Layout Whitepaper*
Version 2.2, 9/29/94
- `/sbin/init.d/template`



Notes

- The whitepaper is located in the "File System Layout" module's "Additional Notes" section.

A. A customer calls to say, "I used to modify our `/etc/rc` file to add the Informix daemons. I just took delivery of my 10.0 system and the old `/etc/rc` file is not there. Where is it nowadays?" Help her out, by giving her the following:

1. A synopsis of why HP-UX changed startup model.
2. A drawing of the new directories, the files in each, and their relationships, which you will fax to her.

3. A brief description of contents of each type of script:

Script	Contents
Execution script	
Configuration variable script	
Start link	
Kill link	
<code>/etc/rc.config</code>	
<code>/sbin/rc</code>	

-
4. List files to modify or create to add daemon, including probable new file names:
-

5. List references (files, documents) to access for examples and more information about new model:

B. She then asks how to determine in what order daemons are being started and killed on the system. Explain the startup/shutdown sequence process:

1. First of all, describe briefly what happens at these run levels, noting examples of processes that are run.

Run level	What happens?	Example processes
0		
S		
1		
2		
3		

-
2. Explain naming convention for start and kill links, by using these examples:

`/sbin/rc2.d/S540sendmail` `/sbin/rc1.d/K460sendmail`

What do the numbers in `rc2.d` and `rc3.d` indicate?

What do S and K as first letters of filenames indicate?

What do numbers in filenames indicate (540, 460)?

What does remainder of each filename indicate (i.e., `sendmail`)?

3. Finally, explain transitioning to run levels:

Which scripts does `/sbin/rc` run when transitioning from a lower run level to a higher one?

What argument does it pass in this case?

What does this argument cause the script to do?

Which scripts does `/sbin/rc` run when transitioning from a higher run level to a lower one?

What argument does it pass in this case?

What does this argument cause the script to do?

1. Here are instructions to install an application on 9.0x. Revise and do for 10.0. (See "Adding startup/shutdown scripts" page in this module for help)

Installation Instructions for TestApp

1. Install software into directory of your choice. This directory, /usr/testapp by default, is called APPDIR in the rest of these instructions.
(STUDENTS: For this lab, we already installed the software in:

/opt/deltaclass/labs/testapp.)

2. Add the startup daemon, \$APPDIR/bin/startup, to the system by editing /etc/rc or /etc/rc.local to add the lines:

```
APPDIR=/usr/testapp/startup
```

```
export APPDIR
```

```
$APPDIR/bin/startup
```

(STUDENTS: For this lab, startup is NOT a daemon. It is just a shell script that echoes a message to console if it executes successfully.)

3. Reboot system, verifying startup program runs.

(STUDENTS: It will echo a message to the console if it is successful.)

2. What does it do to the console display at reboot?
3. Extra credit: Correct the problem, by modifying the executable script (use the 4 functions...)
4. Extra, extra credit: Modify script (see killit script) to "shutdown" TestApp when system is shutdown.

Introduction to Software Distributor



Module goals:

The purpose of this module is to familiarize the students with Software Distributor 2.1 for HP-UX (SD-UX). This includes providing an overview of the SD-UX components, and describing software installation, verification, configuration and removal with SD-UX.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Explain the motivations for changing the software distribution process.
2. Compare DUI and SD-UX, including replaced and new features.
3. Describe the historical basis for SD-UX.
4. Describe and define the components of an SD-UX system.
5. Describe and define the software objects used by SD-UX.
6. List and describe the steps in the software installation process.
7. Use SD-UX to install and manage software.

Motivations for changing tools

- Comply with (and drive!) standards
 - Predecessor to new tool was chosen by OSF and POSIX standards working group
- Support client/server distributed environments
 - *NFS Diskless support in 10.01*
- Make more usable
 - Graphical, terminal and command line user interfaces
 - Similar usage, look and feel to all new utilities

Notes

- Software Distributor 2.1 for HP-UX (SD-UX) was developed to:
 - conform with industry standards for software distribution
 - address the needs of the installed customer base
 - provide a more user-friendly software distribution process
- It is based on HP's already successful new generation of software distribution tools.
 - Information about the history of SD-UX will be covered later in this module.

Features and benefits of SD-UX

Features

- Contains all functionality of previous HP-UX software management tools, and more, such as:
 - Lists, configures and verifies installed software
 - Lists and modifies Access Control Lists (ACLs)
 - Allows multiple versions and source locations

Benefits

- Simplifies system administration by making software distribution faster, more flexible, traceable, and configurable



Notes

- On HP-UX 10.0, SD-UX supports the same concepts and functionality of managing software as in previous releases.
- Additionally, it allows system administrators to
 - look at the features, status and attributes of installed software
 - configure and unconfigure software outside of performing an installation or update
 - verify the correctness of an installed product

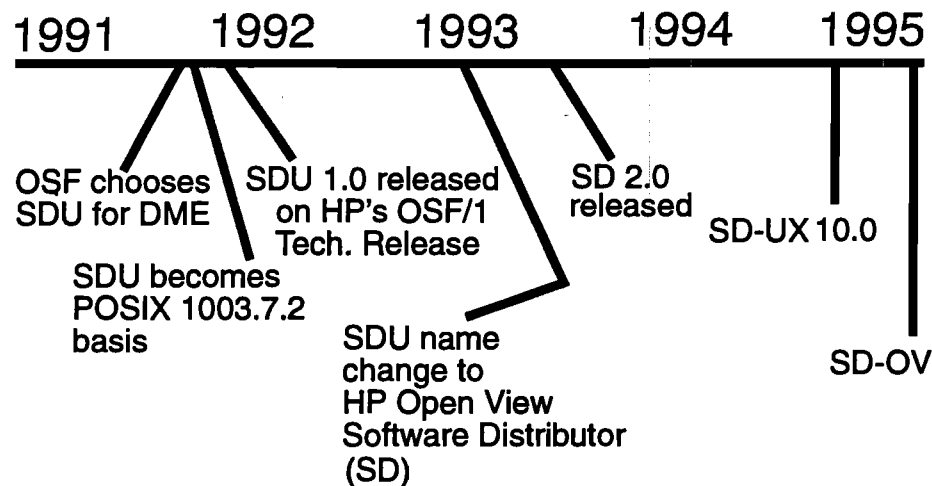
SD-UX vs. HP-UX 9.0x tools

Previous tools	SD-UX equivalent	Purpose
update	swinstall	Install software
update	swinstall	Update existing software
rmfn	swremove	Remove software
fpkg	swpackage	Create media to install
updist	swcopy	Create or update software distribution server structure
netdistd	swagentd	Distribution daemon
netdist server area/ port	depot	Distribution structure
customize scripts	control scripts	Perform auxiliary installation or removal actions

Notes

- In HP-UX 10.0, none of the previous software management tools, labelled DUI by the Lab, the “distributed update and installation tool set”, are available.
 - SD-UX media cannot be read using DUI tools.
 - DUI tools cannot be used with HP-UX 10.0 software.
- SD-UX has a completely different command set, and different terminology and structure, as is described later in this module.
- The above list indicates the SD-UX commands, and other components that replace those used in previous releases of HP-UX.
- SD-UX also provides features and functionality beyond that of HP-UX’s previous software management tools.

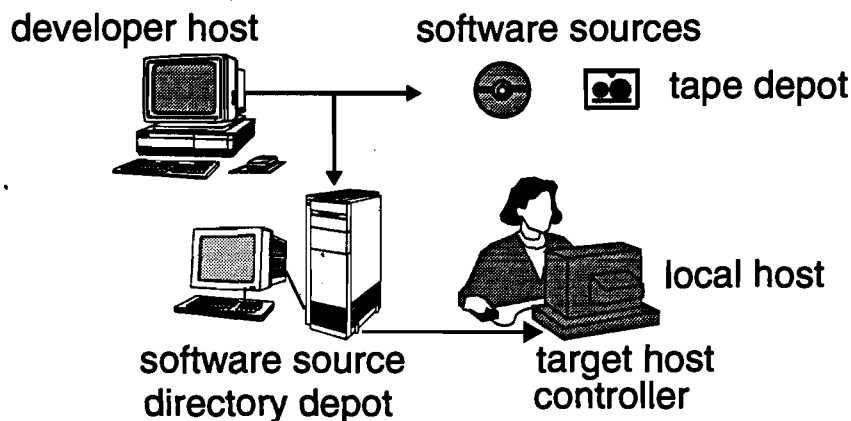
History of SD-UX



Notes

- HP's UDL lab in OSSD developed Software Distribution Utilities (SDU) as next generation software distribution tools.
- In September 1991, OSF chose HP's SDU as the software distribution toolset for the OSF DME (Distributed Management Environment).
- In October 1991, the POSIX 1387.2 (was 1003.7.2) working group selected SDU as basis for their Software Administration standardization work.
 - POSIX 1387.2 describes software distribution standards.
- In 1992, SDU 1.0 was released on HP OSF/1 Technology Release (e.g., Acorn).
- In 1993, the product name was changed to HP Open View Software Distributor (SD).
- In July 1993, SD 2.0 was released.
- SD for HP-UX (SD-UX 10.0) comes with HP-UX 10.0.
- SD for HP OpenView (SD-OV), a purchasable product available after HP-UX 10.0, will be managed as a superset of SD-UX.

SD-UX system overview



Notes

- SD-UX works within a networked environment to distribute software.

Term	Definition
Host	Individual networked computer on which SD-UX operations are performed
Developer host	Where software taken directly from developers is packaged for future distribution
Software source (or) Depot	Physical medium, directory location or host that contains software packaged in SD-UX format: Gathering place for software to be installed by SD-UX
Directory depot	Directory of files and subdirectories organized in SD-UX format. Can be used to install software directly to a target host on network. Can be on local host.
Tape depot	Tape formatted as tar archive; copy to directory depot to use over network
Controller host	Where you invoke SD-UX commands (local host)
Target host	Destination where software is to be installed or managed using SD-UX (local host)

SD-UX daemon/agent

- Every target host and depot host uses these two processes when running SD-UX commands:

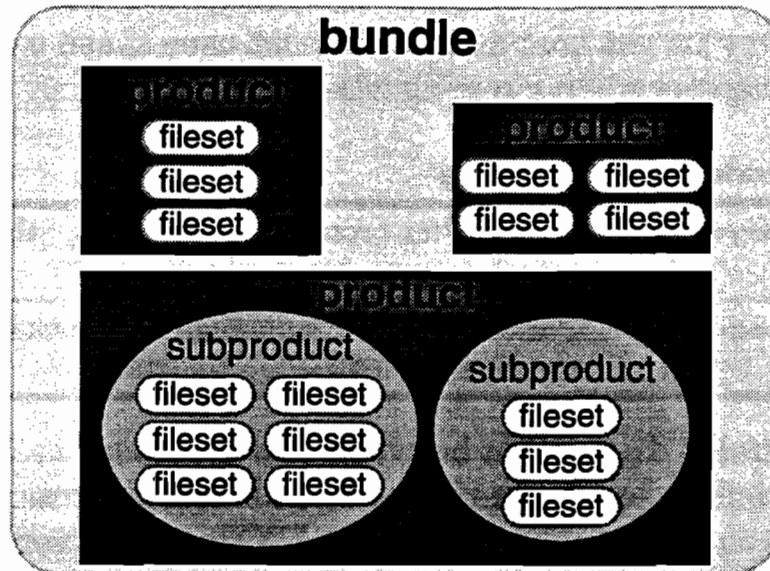
Process	Purpose
swagentd	Listens for controller requests, then schedules agent to do the work
swagent	Performs software management tasks as the agent of an SD-UX command

- Usually, distinction between two processes is invisible, so often referred to as single process:
 - swagentd/swagent or SD-UX daemon/agent.

Notes

- Each system using SD-UX must have swagentd running on it to act as a server for the SD-UX requests.
- The swagent agent process is executed by swagentd, never by the user.
- The swagent is agent that performs software management tasks, as requested by the actual SD-UX command.
 - It does the actual software installation, configuration, verification, etc.
- Refer to man page for these processes.

SD-UX software objects



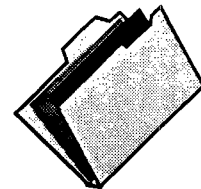
Notes

- Set of software objects selected to manipulate is called software selection.

Term	Definition
Fileset (similar to DUI filesets)	Smallest manageable (selectable) SD-UX software unit Include product files and control scripts that make up a product Can only be a member of a single product
Subproduct	Collection of logically grouped filesets For SD-UX, optional subset of product, needed if many filesets HP-UX software has subproducts, for consistency
Products	All the components of an application Collections of subproducts, filesets and files
Bundle	Grouping of products and/or filesets for a single purpose Treated by SD-UX as a single software entity Represents Corporate Price List (CPL) products; name is partnumber Currently, only HP should make bundles, not customers

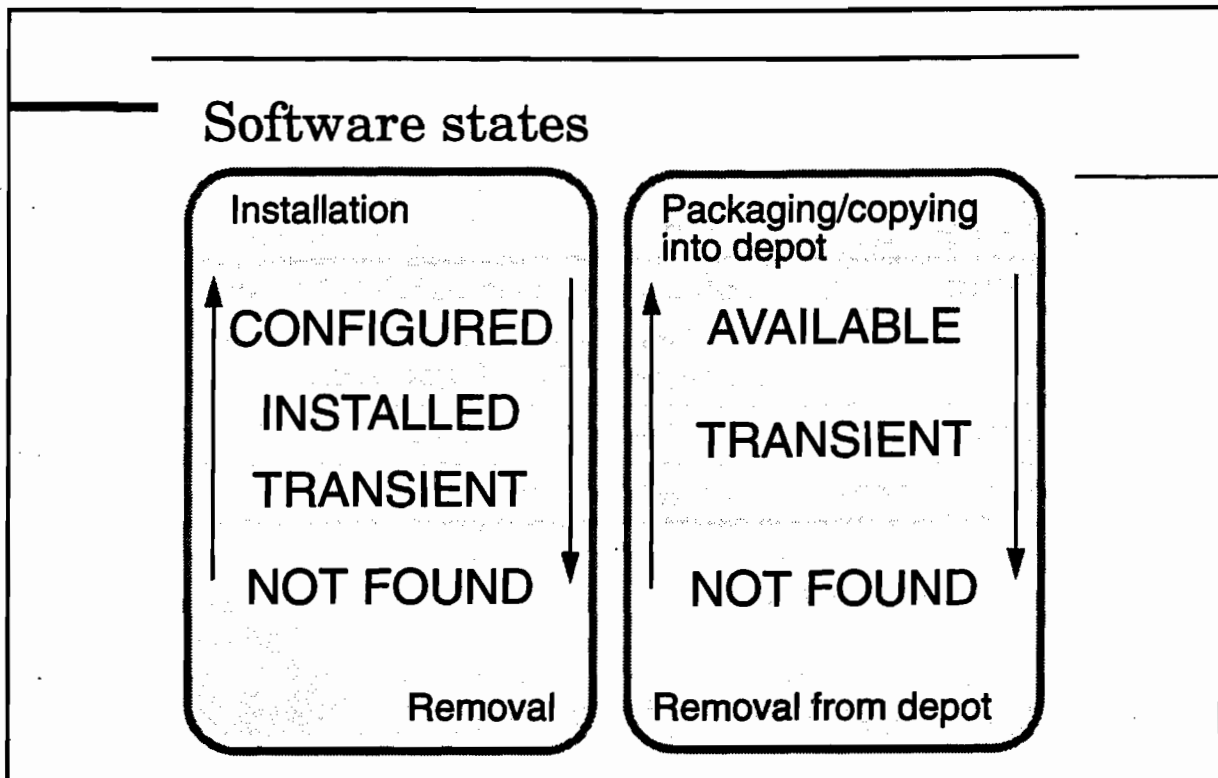
SD-UX input files

File type	File name
Defaults	<code>/usr/lib/sw/sys.defaults</code> (as shipped) <code>/var/adm/sw/defaults</code> (system-wide) <code>~/.sw/defaults</code> (individual user)
Session	<code>~/.sw/sessions</code> (directory)
Software selection	<code>/var/adm/sw/software</code> (directory)



Notes

- These can be manually edited, specified individually on command line and overridden on command line.
- Additionally, they can be modified from the *Options* menu when running SD-UX commands interactively.
- Defaults file contains software management options and default values.
- Refer to `/usr/lib/sw/sys.defaults` for full list of all default values, that come “as shipped,” and a description of their effects.
 - System-wide settings can be configured in `/var/adm/sw/defaults` file, for individual systems.
 - If value not specified here, then `/usr/lib/sw/sys.defaults` is used.
 - Copy `/usr/lib/sw/sys.defaults` to `~/.sw/defaults` for user to modify for running SD-UX tools.
- Current options are saved in session file, just before SD-UX starts a task.
 - After installing, previous session file can be re-used.
- The software selection directory contains files listing names of software selections to be managed.



Notes

- SD-UX commands transition products and filesets through various states.
- Control scripts and SD-UX commands transition products and filesets through various states.
- Here are the states:

State	Description
CONFIGURED	Product/fileset installed and enabled for use
AVAILABLE	Software within depot can be accessed for install by target host
INSTALLED	Software present on target host, but not ready for use
TRANSIENT	Software in process of being installed or removed
CORRUPT	Task failed while software in TRANSIENT state
NOT FOUND	Non-existent, software object not known to SD-UX

Control scripts

- These are optional for any product, and are used to enhance software management tasks
- Successors to customize and decustomize scripts provided for every fileset within previous software distribution environment
- Provide platform for developers to manipulate:
 - installation
 - (un)configuration
 - verification
 - removal
- These scripts, and SD-UX commands can change state of software, indicating its status to SD-UX

Notes

- Software vendors can provide up to nine different scripts to be executed by `swinstall`, `swremove`, `swverify` or `swconfig`.

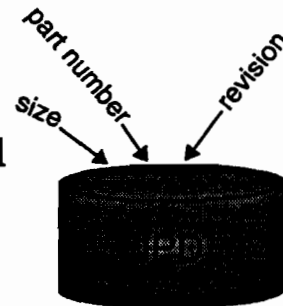
IPD and depot catalogs

Installed Products Database

- Collection of files and subdirectories containing information about all products installed under /.
- Includes software object attributes, such as:
 - product names
 - files and filesets
 - part numbers
 - size
 - vendor
 - revision

Depot catalogs

- Contains information about all products available in depot
- Same structure and types of information as in IPD



Notes

- Each target host has one IPD on it, located by default in /var/adm/sw/products.
- Every software object has a particular set of attributes defined for it in the IPD. These vendor-provided attributes can be displayed using swlist.
 - These attributes are described in the sd(4) man page.
- The IPD is in an HP proprietary format “made up” by HP. It is based on object-oriented techniques, and is not intended to be read or modified directly by customers.
- Manual manipulation can corrupt the IPD and lead to failure of SD-UX commands.
- All viewing and modifications should be made using the SD-UX commands, which automatically query and update the database as part of their functionality.
- Instead of using the IPD, each depot stores information in its own depot catalog, about the software available in the depot. By default, the depot catalog is located in /var/spool/sw/catalog.

SD-UX commands listing

Command	Purpose
<code>swinstall</code>	install and configure software products
<code>swcopy</code>	copy software products to directory depot
<code>swlist*</code>	display information about software products
<code>swverify*</code>	verify software products for compatibility and completeness
<code>swconfig*</code>	configure, unconfigure, or reconfigure installed software
<code>swremove</code>	remove software products
<code>swreg*</code>	control visibility of depots for those performing other SD-UX tasks
<code>swpackage</code>	package software products onto source or for later installation
<code>swmodify*</code>	add, delete and modify attributes and definitions of products in IPD
<code>swacl*</code>	control and manage access to SD-UX objects using ACLs

* Provides additional functionality over DUI

Notes

- Many of these commands support logging, previewing and disk space analysis.
 - Most commands log major events to `/var/adm/sw/<swcommand>.log` on controller, and detailed events to `/var/adm/sw/swagent log` on target host.
 - Previewing makes commands exit after proceeding through analysis phase.
 - Disk space analysis on target host checks for enough space to perform task.
 - For more information about logging, previewing and disk space analysis, refer to `sd(5)` man page.
 - Additionally, see this module's "Additional Notes" section.
- This module describes all commands except for `swpackage` and `swmodify` which are covered in the "Software Distributor Packaging" module.
- Additionally, each command has its own man page.

The swinstall command syntax

```
swinstall [XToolkit Options] [-i] [-l] [-p] [-v] [-r]  
[-s source] [-x option=value] [-Xoption_file]  
[-f file] [-t target_file] [-S session_file]  
[software_selections] [@ target_selections]
```

Option	Description
-v	verbose mode (default)
-s source	specifies software source
-X option_file	use this option_file to override defaults file
-x option=value	set session default option to value
-f file	read software selections from software selection file
-S session_file	use this session_file

Notes

- The above example shows syntax standard for many SD-UX commands.
- The options, which override defaults, are available to most commands.
 - Many commands have additional options, indicated in their man page.
- With -s, specify hostname, hostname alias, or full network address.
 - Optionally, follow by alternate depot: hphost1:/alternate_dir
 - Default source is default depot on the local host: /var/spool/sw.
- Use -f to indicate a file which lists software selections.
 - ASCII text file with each software selection on a separate line, in format for software_selections operand as indicated on next page.
 - Blank lines and lines beginning with # (comment) ignored
 - Software selections in file added to command line software selections
- Use -S to run command based on values saved from previous session.

Listing information: swlist

- Lists attributes (from IPD) for software objects
- Default is to list names of products installed at /

Option	Will list:
<i>-a attribute</i>	<i>specific attribute</i>
<i>-v</i>	<i>all attributes</i>
<i>software_selection</i>	<i>specified objects</i>
<i>-l level</i>	<i>all objects down to level</i>
<i>-d depot</i>	<i>objects on depot</i>

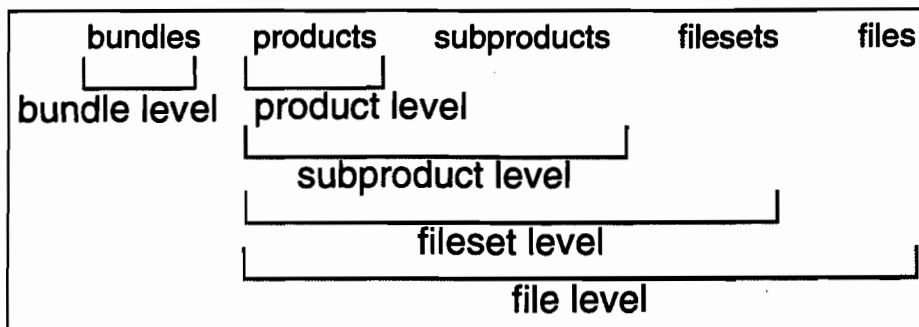
- Additionally, can create list of software objects to use as input for `swinstall` or `swremove`

Notes

- If no options are specified, the default behavior is to display the bundle names only for those installed in the primary root directory.
- Using *-a attribute* will display the specified attribute(s) for the products, in addition to the product name, which is always displayed.
 - Example: `swlist -a title size`
 - Refer to `sd(4)` man for a listing of attributes.
- Set the defaults file's `one_liner` option to the desired attributes to display instead of using command line. Command line overrides.
- Using *-v* will display all attributes for the products.
- To list a specific software object, use *software_selection*.
 - Example: `swlist -v Networking`
- Combine options to further customize the output.
- Output from `swlist` can be contents for `swinstall/swremove -f file`.
- An example of `swlist` output is in this module's "Additional Notes" section.

Listing levels

- Listing levels determine what objects are listed, in conjunction with `software_selection`.



- In addition to above levels, the level depot displays attributes of depots on specified host.

Notes

- Specifying a level will list information about the objects at that level in addition to all those that are above the level.
 - The exception is the depot level, which only lists information about the actual depot, as opposed to its contents.
 - A depot's contents can be listed by using the `-d` option.
- Each software object has a particular set of attributes. By default, only the names of the software objects at the specified level and above will display.
 - Refer to `sd(4)` man page for attributes list for each software object type.
- Use `-v` with `-l` to display all attributes for a particular level.
 - Example: `swlist -vl fileset Networking`
 - Use `-a attribute` with `-l` to display particular attributes for that level and above.
 - Example: `swlist -l fileset -a state -a size Networking`
- Set the defaults file's `level` option to the desired level to display instead of using command line. Override with `-l` option.
- See "Additional Notes" for examples of running `swlist`.

Installing software: swinstall

- Installs software from depot on local host to local host's root filesystem, and configures for use

Option	Will install:
<i>-s source</i>	from specified <i>source</i>
<i>software_selection</i>	specified objects only

- Checks for compatibility and dependencies
- Supports kernel rebuilding and rebooting
- Can install multiple versions of software
- Can compress files over network

Notes

- Default is to install all software located in /var/spool/sw to /. It then prepares the software, bringing it to the CONFIGURED state.
 - A reboot may be required before the software is configured.
- By default, it will only install higher revisions of software.
 - Override this by setting defaults file options `swinstall.reinstall` and `swinstall.allow_downdate` to true.
- By default, files are removed as they are updated.
 - Open binaries are renamed before updating and renamed files are removed after reboot.
- If load of fileset fails, then state is CORRUPT, and install must be re-tried.
 - To create backup copies which are removed only after successful install, set `swinstall.autorecover_product` to true.
 - This allows recovery or rollback to original product.
- Setting the defaults file's `compress_files` option to true will reduce network usage by approximately 50%, and thereby increase performance.
 - Only do when network bandwidth clearly restricts total throughput.

Compatibilities and dependencies

Compatible software

- Software products specify with which machine type(s) and OS(s) they are compatible
- Some SD-UX commands can detect and/or enforce use of compatible software

Dependent software

- SD-UX supports two types of dependencies
 - Pre-requisite: install and configure before dependent fileset is installed and configured
 - Co-requisite: install and configure before dependent fileset is usable
- Default is to resolve all before command proceeds

Notes

- A depot may contain products for a variety of machines or architectures.
- The `swinstall` command performs compatibility filtering to ensure architecture of products matches host when software is installed to default target directory (/).
 - It uses the `uname` attribute for this. Refer to `sd(4)` man page for more information about this and other attributes.
 - To change this default, set `allow_incompatible` to true.
- By default, the `swinstall` and `swcopy` commands support dependencies.
 - To not enforce dependencies, set default option `enforce_dependencies` to false.
- If the software selection specifies dependencies, the commands will automatically select that software.
 - To not do this, set `autoselect_dependencies` to false.

Product location, multiple versions

Alternate product location

- Software can be installed to location other than vendor-specified default
- Install location becomes root directory of all product's files

Multiple software versions

- Multiple versions of software product can be installed to single target system, each in unique product location
- User can select specific product from multiple installed versions by specifying product location as part of software selection

Notes

- If the software supports it, SD-UX allows having multiple versions of the same software on:
 - different hosts
 - different directories of the same host
 - same directory of one host
- But only one version can be CONFIGURED per system.
- Multiple versions may be necessary because users:
 - may not be using the same version
 - may not be using the same platform
 - may require different configurations of the software
- To allow multiple version support, set the defaults file option `allow_multiple_versions` to true.

Kernel rebuilding and rebooting

Kernel rebuilding

- After software which modifies kernel is installed or updated, `swinstall` executes system-specific command to create new version of kernel

Rebooting

- After software which requires a system reboot is installed, `swinstall` automatically reboots
 - If software attribute `is_reboot` set to `true` this requires a reboot
 - Partially controlled by `autoreboot` default option also

Notes

- When updating HP-UX OS to a new revision:
 1. Kernel software is installed first.
 2. New kernel is generated before updating rest of OS.
 3. After rest of OS is updated, `swinstall` reboots using new kernel.
 - No "reconnect" after reboot
- All software which modifies the kernel will also require a reboot.
 - The `is_reboot` attribute must be set for each `is_kernel` attribute.
- In comparison to DUI, SD-UX accomplishes an OS update with only one system reboot, compared to DUI's two.

Phases of installation

Phase	Purpose
Selection	Specify target, source and software selection
Analysis	Check for free disk space, dependencies, compatibilities, mounted volumes, etc.
Load	Load software on target, replacing old software in process
Configuration	Kernel rebuild and reboot, if needed, modify startup/shutdown scripts, provide info to administrator

Notes

- Each phase has certain tasks performed by swinstall, or by the control scripts which swinstall runs at various times.
- The selection phase can be based on defaults:
 - target is the local host's /
 - source is the local host's /var/spool/sw
- The other phases can all include tasks performed by control scripts.
- These control scripts are discussed later in this module.
- The analysis phase is described on the next page.

Results of analysis

Result	Meaning
Ready	No problems — install proceeds
Ready with Warnings	Some warnings—see log file
Ready with Errors	At least one product will be installed— one or more are excluded due to analysis errors, see log file
Communication failure	Contact or communication with intended target or source has been lost
Excluded due to errors	Global error occurred, preventing install
Disk Space Failure	Installation will exceed free disk space on target—see <i>Disk Space</i>

Notes

- The analysis phase proceeds through various stages, ending with one of the above results.
 - Analyzing targets
 - Analyzing software
 - Finishing analysis
 - Ready
- When analysis is done, target host's final status will be "Ready" or "Excluded from task"
- Select *OK* to proceed to installation phase, if host is "Ready".

Control scripts and swinstall

Script	Purpose
checkinstall	Tests for hardware and software configurations that might prevent install
preinstall	Cleans up from predecessor files—removes obsolete files, kills daemons owned by fileset
postinstall	Enhances kernel functionality—adds drivers, modifies kernel parameters
configure	Creates any special files, performs conditional moves of delivered files, modifies system configuration data files, warns/informs administrator as necessary

Notes

1. The checkinstall script is executed before swinstall performs its own checks (e.g., disk space analysis, compatibility, dependencies, mounted volumes).
 - If error from a fileset's checkinstall script, fileset is deselected.
 - If error from a product's checkinstall script, product is deselected.
 - If error from other checks, install fails.
2. The preinstall script is run on each fileset, after all checks for that fileset are passed.
3. After loading the fileset, the postinstall script is run, if kernel needs re-configured.
 - NOTE: Steps 2 and 3 are repeated for each fileset, one by one.
4. If error from kernel fileset preinstall or postinstall script, load error, or kernel build failure, install fails.
5. The configure script is executed after the kernel is rebuilt and rebooted.
 - This script is most like the 9.0x's customize script.

Verifying software: swverify

- By default, verifies all INSTALLED or CONFIGURED products in local root's /
- Use these options to change defaults

Option	Purpose:
-d <i>source</i>	verify AVAILABLE software on depot
-f <i>file</i>	verify software selections in <i>file</i>
<i>software_selection</i>	verify specified objects only
@ <i>target_selections</i>	verify software on specified target

Notes

- **Example:**
To verify AVAILABLE fileset productx.fileseta on local host's default depot, enter this command:
swverify -d softwarex.fileseta
- The verification process has two phases:
 - Selection of host and software
 - Analysis, which involves the actual verification
- Selection is done by the user indicating host and software in command line, or using the defaults.
- The analysis phase is described on the next page.

Verification—analysis phase

1. Initiate analysis.
2. Process software selections.
 - Access IPD or depot catalog.
 - Check compatibilities, for installed software.
3. Check states of versions.
 - Verify states of filesets are correct
 - Check for multiple versions.
4. Check dependencies.
5. Execute verify control scripts.
 - For CONFIGURED software only
6. Perform file level checks.
 - Such as permissions and checksums



Notes

- Verify control scripts ensure the configuration was performed correctly. Some functions they may perform include:
 - test for existence of files and directories, including IPD
 - read contents and attributes of files and directories
 - verifying licenses
- File level checks include
 - checking for missing files
 - checking for proper symlink values
 - checking for altered files by comparing checksums
 - checking for attributes
 - mode
 - owner
 - group

Removing software: swremove

- Removes specified software installed in local host's /
- Can remove multiple versions
- Can preview removal (-p option)
- Use other standard options to control removal
- Use its GUI, which is similar to that of swinstall and swcopy
- It executes several control scripts



Notes

- The swremove GUI session has three phases:
- Selection
 - Similar to swinstall's selection phase
- Analysis
 - Similar to swinstall's, but no disk space analysis or compatibility checking is done
 - Additionally, for dependencies, swremove will remove the dependents, not the dependencies
- Removal
 - Actually unconfigures and removes the software from the system, and from the IPD or depot catalog
 - Software will be "non-existent"

Control scripts used by swremove

Script	Purpose
checkremove	checks for existence or absence of files, checks hardware/kernel/system configuration
unconfigure	undoes what configure script does, kills processes from previous fileset, removes client-specific files, such as log files
preremove	move or remove files and directories under shared directories; sever symbolic links from a shared directory to another shared dir.
postremove	remove newly emptied directories which are exclusive property of fileset and reside in a shared directory; remove fileset's log files

Notes

- The checkremove script verifies there are no obstacles to removal at the product or fileset level.
- Executing the unconfigure script will change the software state from CONFIGURED to INSTALLED
- Neither the preremove nor the postremove script should perform any of the unconfiguration tasks.
- The postremove script will perform any other actions that bring the system closer to the condition it was in prior to the fileset's installation.

Configuring software: swconfig

- By default, configures all software INSTALLED in local host's /
- Can also unconfigure software (-u)
- Can preview (un)configuration (-p)
- If configuration delayed at installation, by setting defer_configure to true, use swconfig
 - This is done when installing multiple versions
- Command line only

Notes

- Configures host to run software, and configures software for host-specific information
- Generally, an installed product is configured by swinstall. But, in some cases, as described above, swinstall does not perform a configuration.
- Use swconfig to perform same tasks that swinstall does when it configures software, including running the configuration control script.

Phase	Description
Selection	Local host resolves list of software to configure (same as other commands)
Analysis	Determines if the configuration can take place
Configuration	Actual (un)configuration takes place The configure or unconfigure scripts are executed.

Configuration—analysis phase

1. Initiate analysis.
2. Process software selections.
 - Access IPD.
 - Check compatibilities, for installed software.
3. Check states of versions.
 - Verify states of filesets are correct
4. If second version exists:
 - `allow_multiple_versions=true`,
(un)configures a second version
 - `allow_multiple_versions=false`,
(un)configuration fails
5. Check states of dependencies.



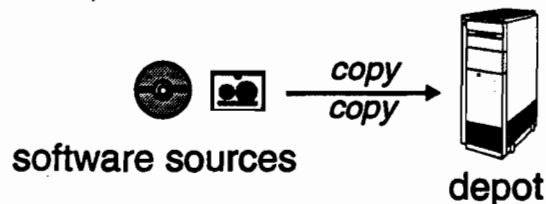
Notes

- For step #3:
 - if product is non-existent or CORRUPT, configuration fails.
 - If already (UN)CONFIGURED, warning occurs.
- During the (un)configuration phase:
 - The product's state changes from INSTALLED to CONFIGURED if being configured.
 - The product's state changes from CONFIGURED to INSTALLED if being unconfigured.

Copying a depot: swcopy

- Copies or merges software from a depot to a directory depot

Option	Will install:
<code>-s source</code>	from specified <i>source</i>
<code>software_selection</code>	specified objects only



Notes

- Copying is done to create new depots for later installation. Contents of depot can be accessed as a software source by `swinstall`.
 - The software is put into a place, made **AVAILABLE** there, but not **INSTALLED**.
 - No control scripts are run, and the kernel does not need to be rebuilt.
 - In addition, compatibility checking is not done, because the software depot might be on one system for installation onto another.
- The `swcopy` command can copy to directory depots only, not tape depots.
 - Only `swpackage` can create or modify tape depot contents
- When using the command line, `swcopy` supports the preview mode, which will run the session through an analysis phase only.
 - The analysis phase, and other phases, will be described in the `swinstall` section of this module.
- The `swcopy` command has a GUI, which is highly recommended.

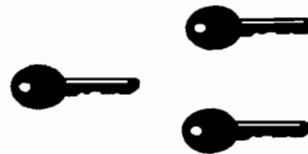
Registering depots: swreg

- By default, registers default depot on local host
 - Registration makes a depot usable as a source
- Can unregister a depot also
- By default, swcopy registers newly created depots
- By default, swremove unregisters a depot when it removes all the software from it
- Use swreg for cases such as:
 - Making CD-ROM available as depot
 - Making depot available that swpackage created
 - Making depot unavailable without physically removing it

Notes

Controlling access with ACLs

- SD-UX provides an Access Control List (ACL) for:
 - SD-UX hosts
 - directory depots
 - root filesystems
 - software products within depots
- An object's ACL provides a set of "keys" which can permit or prevent individuals from performing specific operations on the object, such as:
 - viewing or editing its ACL
 - installing a new product
 - listing its attributes
 - deleting a product



Notes

- These are not the same as the HP-UX ACLs.
- These extend traditional HP-UX file level permissions.

Example ACL: for root

```
#
# swacl      Installed Software Access Control List
#
# For host: hpcusp01:/
#
# Date: Fri Oct 14 20:20:48 1994
#
# Object Ownership:  User= root
#                    Group= sys
#                    Realm=hpcusp01
#
# default_realm=hpcusp01
object_owner:crwit
group:swadm:crwit
any_other:-r--t
```

Notes

- Object ownership:
 - User = object_owner
 - Group = object_group
 - Realm = hostname of system where object is located

ACL levels

-1 level designation	View/modify ACL protecting:
host	host(s) specified by <i>target_selections</i>
root	root filesystem(s) specified by <i>target_selections</i>
depot	depot(s) specified by <i>target_selections</i>
product	products specified by <i>software_selection</i>
product_template (product ACL template)	template for ACLs of future products added to depots specified by <i>target_selections</i>
global_soc_template (container ACL template)	... of future depots or root filesystems added to hosts specified by <i>target_selections</i>
global_product_template (product ACL template)	... of product template ACLs of future depots added to hosts identified by <i>target_selections</i>

Notes

- The *-1 level* specifies which level of SD-UX ACLs to view or modify.
- Templates are used to initialize ACLs of new objects, to set up default parameters.
- The product level is only for products in a depot. Products installed in a root filesystem are protected by the root's ACL.
- The highest level of protected object is the host system.

ACL protecting:	Controls permission to:
host	create depots and roots on the host
depot	manage depot and add new products to it
root	manage installed products in the root
product	administer product in depot, including installing to a target
any template	view or modify template

ACL entries

- Consist of three fields:

entry_type[:key]:permission

Field	Content
<i>entry_type</i>	type who/that gets permission(s) (e.g., user, group, host)
<i>:key</i>	actual name of user, host, etc. getting permission (e.g., cindy, hphost1)
<i>:permissions</i>	type of access given

- Example: user:chris:r--t-

Notes

- The table below lists and describes accepted entry types and key values:

Entry type	Permissions apply to:	Key value
object_owner	user who created object	---
user	named user	user name
object_group	members of group to which object belongs	---
group	members of named group	group name
host	SD-UX agent from named host	host name or Internet number (dot designation)
other	groups or users not named	no name
any_other	users or hosts not named	no name

ACL permissions

	host system	root	depot	product on depot
control	edit ACL for object			
test	test access to object, read (list) ACL for object			
insert	insert/install depot/root	insert/install product	insert/install product	n/a
write	change host	change or delete root or products	change or delete depot	change or delete product
read	list depots and roots	list root and product attributes	list depot and product attributes	read product files

Notes

- In the ACL entry, these permissions are abbreviated to c, t, i, w, and r.
- If all permissions are granted, use a as shorthand for listing them all.
- Local superusers are automatically granted all permissions, regardless of ACLs.
- Other users are granted permissions by looking at the most specific to the most general *entry_type* (e.g., *object_owner*, then *user...*).
 - For example, even if *user:chris* has more restrictive permissions than *other*, the more restrictive permissions of *user:chris* apply to *chris*.
- Each task has specific permission requirements. These are discussed in detail for each SD-UX command and task in *Managing HP-UX Software*, Chapter 7, "Controlling Access to Software Objects."
 - See list of permissions for *swinstall* in this module's "Additional Notes" section.

Controlling access using swacl

- By default, displays ACL entries for specified object

Option	Purpose:
-M <i>acl_entry</i>	Add or change <i>acl_entry</i>
-D <i>acl_entry</i>	Delete <i>acl_entry</i>
-F <i>acl_file</i>	Overwrite ACL with <i>acl_file</i>
-l <i>level</i>	Edit specified object <i>level</i> (NOT swlist -l <i>level</i>)

- No GUI, so use command line options above or standard command options listed previously

Notes

- The swacl command can be used to view or edit ACLs.
- To use the swacl command, the user must have the appropriate permissions, assigned within the object's ACL, for the specified object.
- See this module's "Additional Notes" section for examples of using swacl.
- NOTE: Regardless of the ACL, a user can know of a depot's *existence* by using swlist -l depot.

Tips and tricks

- SD-UX manages itself like any other product, **EXCEPT** it cannot remove itself.
- If a SD-UX command cannot be run because a software object's lock has not been removed, remove the lock manually.
 - To remove an unwanted lock, remove the swlock file located in `/var/adm/sw/product`.



Notes

- The swlock is used to prevent simultaneous alterations to a software object.
- It may hold an unwanted lock on the local host's / for example, if root had been doing a swinstall while remotely logged in, and there was a network problem.

Summary and review

- SD-UX, a standards-compliant set of tools, replaces previous software management tools.
- Allows managing bundles of software including providing the ability to install and configure:
 - HP-UX OS
 - multiple versions of software
- Controls security through ACLs
- Allows software management customization by vendor-provided control scripts
- Stores software attributes in IPD and catalogs

Notes

- Of course there are many other new features provided by SD-UX, including:
 - a TUI for S800 systems
 - the ability to list the software attributes in IPD and the depot catalogs
 - verification of the integrity of installed and configured software
 - the ability to control the behavior of various SD-UX commands via a hierarchy of defaults options files
 - a daemon/agent that acts as server/agent for performing SD-UX tasks

Co-existence with HP-UX 9.0

- There is a utility that converts the 9.0x fpkg format to swpackage format: fpkg2swpkg
 - This utility is discussed in the next module.
- No DUI utilities are supported for running on HP-UX 10.0.

Notes

Syntax: software_selections

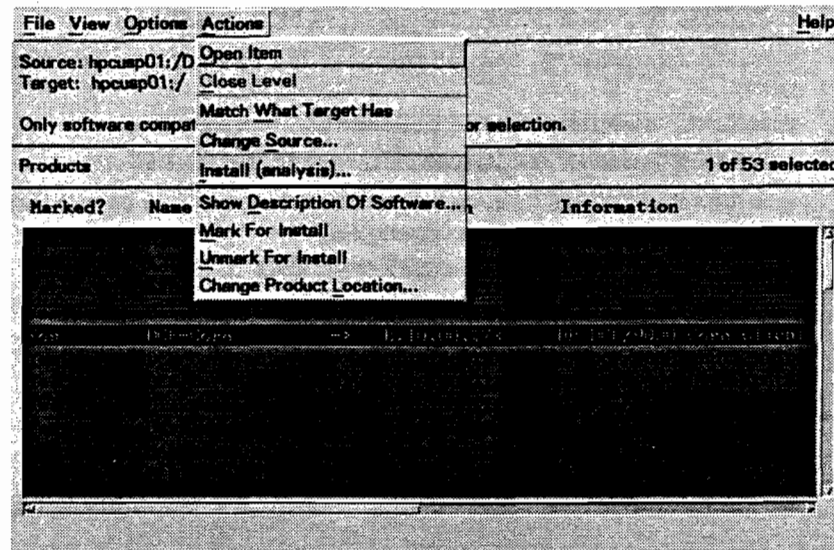
- **Syntax for software_selections:**
bundle[.product[.subproduct][.fileset][,version]]
or
product[.subproduct][.fileset][,version]
- **The version component is specified as:**
[,r<op>revision][,a<op>architecture][,v<op>vendor]
[,c<op>category][,instance_id]
where <op> is a relational operator and
instance_id is number to further distinguish versions
 - Relational operators force individual comparisons on dot-separated fields.
- **Examples of software_selections:**
 - 1 - SoftwareA.subproductAa.filesetAA,r<C,a=HP-UX
 - 2 - BundleX
 - 3 - ProductY,4

Notes

- Software objects can be described very precisely or very broadly.
 - Minimally, the bundle name or product name must be referenced.
 - At the other extreme, an object can be described very hierarchically. If so, the software objects are separated by a period.
- If the `software_selection` specifies a bundle, product or subproduct, all of that object's filesets will be included.
- Version specifications (r,a,v,c) are repeatable and separated by a comma.
- Relational operators are =, ==, >=, <=, <, >, or != .
 - Using any relational operator but for = (equals) will cause system to compare each dot-separated field to find matches.
 - Example: `r>=B.10.00`
indicates to choose all revisions greater than or equal to B.10.0
 - When matches within each field are satisfied, software will be selected.
 - Only = (equals) provides shell pattern matching notations or wildcards.
 - No isspace(3) characters are allowed.
- Only use `instance_id` when certain of the product contents.

- Version 1.1.1 Module 7, page 141

“Software Selection Window”



Notes

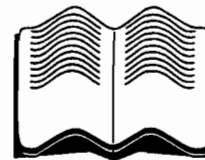
- The `swinstall`, `swremove` and `swcopy` commands can be run interactively, using a GUI or TUI (terminal user interface).
 - They are accessible from SAM.
- Use GUI or TUI as they are fast, easy to use, and help user with problems.
- GUI interface is Motif, and can be modified with Xtoolkit command line options.
- There is a menubar, a message area, and an objects list.
- There is on-line help, and an options editor for modifying defaults.
- The main menu bar has five options:
 1. *File* - manage session files
 2. *View* - change how data is presented
 3. *Options* - change SD-UX options
 4. *Actions* - select targets and software, and perform actions
 5. *Help* - helpful information and on line documentation
- Options vary depending on which task is being performed.

For more information



For more information

- *Managing HP-UX Software*
- The man pages for `sd(4)` and `sd(5)`
- The man pages for each command



Notes

- References for packaging with Software Distributor will be covered in the "Software Distributor Packaging" module.

A customer calls you with several questions about how to install and removing HP-UX software. She cannot find `update` or `rmfn` on her system. You tell her that these tools have been replaced by a new set of software management tools, called Software Distributor. Help her get used to SD-UX by answering the following questions:

1. Why did HP switch to SD-UX? (Briefly explain where SD-UX came from, and what standards it supports).
2. Draw your own picture showing the hierarchy of software objects: (to fax to her):
3. Describe the purpose of `swagent` and `swagentd`.
4. Draw filesystem hierarchy showing the 3 types of SD-UX input files, and describe purpose of each: (to fax)

5. Complete the following table, to fax to her, also:

SD-UX command	Purpose	Replaces (state if a new feature)
swinstall		
swcopy		
swlist		
swverify		
swconfig		
swremove		
swreg		
swpackage		
swmodify		
swacl		

In this lab, you will become familiar with SD-UX's man pages, features and commands. You will look at the installed software, remove some, re-install it, verify it and look at and modify access control lists.

1. Compare `sd(4)` man page with `sd(5)` man page.
What is the purpose of each?

2. Refer to appropriate man page to answer these:
How does `-p` option help an SD-UX user?

Where are log files stored for each command?

Which signals cause SD-UX agent to exit?

What problems may occur if agent is killed?

What is purpose of `/var/adm/sw/host_object` file?

3. Verify the SD-UX daemon is running on the system:
`ps -ef | grep swagentd`

Use its man page to answer the following questions:
What does option setting `max_agent=-1` mean?

Can you manually start the daemon? _____
If so, how?

-
4. Look at the software installed on the system:

`swlist`

What is SD-UX's product name? _____

List filesets in SD-UX.

Which option do you use to do this? _____

What are the filesets?

List files in SD-UX manual pages.

Which option do you use to do this? _____

List size of SD-UX product and filesets.

Which option do you use to do this? _____

How big is the SD-UX manual? _____

Look at README file for the SD-UX product.

List verbosely SD-UX's product attributes.

What are some of the additional things viewable now?

List information about your depot.

What option did you use? _____

5. Set up and modify your own SD-UX defaults file, by
copying SD-UX system defaults file to your `~/.sw`:
`cp /usr/lib/sw/sys.defaults ~/.sw/defaults`

Open your own defaults file with an editor.

What is `swlist.one_liner`'s purpose?

Change its value to `= size title`.

Save the file and exit the editor.

-
6. List all products on the system again.
What additional attributes are listed by default now? _____

-
7. Try to remove some software from the system:

swremove

Set the view to *Products* (not *Bundles*).

Double-click *DCE-Core* product to "open" it.

Open *Release Notes*, then *DCE-CORE-NOTES*.

View *DCE-CORE_NOTES' Description*. List its files.

Look at description of *DCE-Core*.

What are some of its dependencies?

Mark *DCE-Core* for removal.

What happens?

What is the problem?

Try to remove *DCE-Core* anyway. What happens?

(Refer to the on-line help, if needed.)

Look at the *Logfile*.

What are some products that require *DCE-Core*?

Cancel removal process, but do not exit swremove.

What would you do to allow removal anyway?
(Hint: *Options...*)

Do not remove *DCE-Core*, but do this:
Analyze removal with *Enforce dependency analysis errors...* option set to *False*.

What does the logfile tell you this time?

Cancel removal process.

Select *File...Clear Session* and continue the lab.

8. Now, you can really remove software:
Remove entire *Keyshell* product and only the
Icelandic fileset of *International* product.

After removal, look at *Product Summary* from
"Remove Window".

What is *International* product marked as now?

Look at log file. What phases occur after analysis?

What is the *Product Description*?

Exit swremove.

Can you run keysh?_____

-
9. Re-install *Keyshell* and *Icelandic*, by running:
swinstall

Use the depot from which you used to install HP-UX.

Ensure *Software View* is for *Products*.

Mark these two for install, and do the Install analysis.

Look at *Disk Space* and *Product Summary*.

What does *Skip as Projected Action* mean?

Refer to logfile.

Complete Installation, then view *Product Summary*
before exiting.

What is the status of *Keyshell* and *International* now?

In log file: What phases occurred after analysis?

Exit swinstall.

Does keysh work now?

10. Run swverify on *Keyshell*.
View the swverify log file after it has completed.
(Refer to its man page for more information, including
location of its log file.)
To which file does the swverify log file refer you?

-
11. View and modify access control lists on the system.
Refer to this module, and the swacl man page.

View the ACL of the root file system. What command?

Can you view the ACL of a product installed on your
depot, such as SD-UX (product SW-DIST)?

Modify root file system's ACL so that only superuser can list its ACL and product attributes. Command? _____

Test as a non-super user, using `swlist` and `swacl`.
What do the errors tell you?

Modify the ACL back to its original state, and re-test.

12. Extra credit:

Create a software selection file, and use it with SD-UX commands, such as: `swlist -f filename`

13. Extra credit:

Run `swconfig`. To do this:

Remove *Keyshell* again, and re-install it, but set the option (GUT's Options..., or defaults option) "defer configuration" to true.

After installation, view *Product Summary* and logfile.

What is Keyshell's status? _____

Run `swconfig` on Keyshell.

View `swagent.log` for details.

This page intentionally left blank.

Additional notes for “Introduction to SD-UX” module

This page intentionally left blank.

SD-UX commands comparison

Command	Logging?	Previewing?	Disk-space analysis?	GUI/TUI?
swinstall	Yes	Yes	Yes, always	Yes
swcopy	Yes	Yes	Yes, always	Yes
swlist	No	No	No	No
swverify	Yes	No	No	No
swconfig	Yes	Yes	No	No
swremove	Yes	Yes	Optionally	Yes
swreg	Yes	No	No	No
swpackage	Yes	Yes	Yes, always	No
swmodify	Yes	No	No	No
swacl	Yes	No	No	No

Notes

- Most commands log major events to `/var/adm/sw/<swcommand>.log` and detailed events to `/var/adm/sw/swagent.log`.
 - Major events, such as beginning and end of task, also go to `stdout`.
 - Errors, such as the host failing analysis or failing a task, go to `stderr`.
 - The `swpackage` command logs all event to the same log file.
- Previewing makes commands exit after proceeding through analysis phase.
 - The `swpackage` command goes beyond analysis, but makes no changes to the destination depot.
 - GUI or TUI waits for confirmation after analysis and will show invalid conditions on screen, so user can determine whether or not to proceed.
- Disk space analysis on target host checks for enough space to perform task.
 - Set defaults option `enforce_dsa` to `false`, to prevent enforcing lack of disk space by stopping task.
 - Instead, operation will use filesystem's `minfree` space, which fails if reaches filesystem's absolute limit.
 - Can see detailed disk space information in GUI/TUI.

Listing media attributes

- List media attributes of local tape depot:
`swlist -v -l depot @ /dev/rmt/0m`
or
`swlist -vl depot -s /dev/rmt/0m`
- Outputs this listing:

type	distribution
tag	CORE OS
description	HP-UX Core Operating System...
number	B2358-13601
date	June 1991

Notes

Example of using swacl

- To add an entry for user ann to ACL for COBOL:
`swacl -l product -M user:ann:cr--t COBOL`
- To delete entry for user john from ACL protecting FORTRAN:
`swacl -l product -D user:john FORTRAN`
- To list the ACL protecting FORTRAN:
`swacl -l product FORTRAN`

Notes

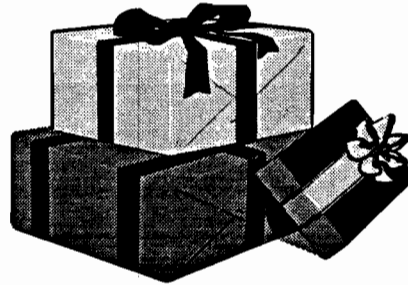
Permissions for using swinstall

1. Agent verifies user can insert on target root.
2. Agent verifies user can write on target root.
3. Agent for source verifies system can read source product.
4. Agent for source verifies depot is registered. If not, agent verifies controller user and target host both can insert on source's host.

Notes

- Each task has specific permission requirements.
- These are discussed in detail for each SD-UX command and task in *Managing HP-UX Software*, Chapter 7, "Controlling Access to Software Objects."

Software Distributor Packaging



Module goals:

The purpose of this module is to learn how to create a Software Distributor (SD-UX) package for installation.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the components of a Software Distributor (SD-UX) package.
2. Create a Software Distributor (SD-UX) package, including creating a Product Specification File (PSF) and editing control scripts.
3. Use `swmodify` to alter a SD-UX package.
4. Use `fpkg2swpkg` to convert an `fpkg` PSF or `netdist` to `swpackage` PSF.

Software Distributor package

executables, configuration files

+

control scripts

=

+

Product Specification File



- The package creator must determine:
 - Files and directories to include to run product
 - Product attributes
 - Product structure
 - Need for and function of control scripts

Notes

- Software Distributor (SD-UX) package consists of:
 - Files required to run the product
 - executables
 - configuration files
 - Control scripts
 - customize product's installation, configuration, verification and removal
 - Product Specification File (PSF)
 - describes product's structure, attributes, contents, compatibilities and dependencies

Product Specification File (PSF)

- Drives the swpackage session
- Defines product structure, attributes, contents
- Specifies attributes for software objects:
 - One or more products or bundles
 - Only HP can create bundles
 - Subproducts for product
 - Required by HP, not SD-UX
 - Filesets for subproducts
- Optionally, can specify attributes for:
 - Target depot/tape
 - Vendor



Notes

- The Product Specification File, or PSF, is used to describe the product or bundle in detail.
- In it, the packager developer can provide any or all of the available attributes.
- Refer to *sd(4)* man page or *Managing HP-UX Software*, Ch. 8, "Packaging Overview" for a complete list of attributes.

PSF attributes for all objects

- All objects are required to have the following:

Attribute	Description	Size
tag	short descriptive name or identifier for object	16 chars max usually

- All objects are recommended to have these:

Attribute	Description	Size
title	more detailed name for object, chance to expand on tag	80 chars max usually
description	a file providing a detailed description of the object	up to 8K usually

Notes

- All objects from HP-UX software teams are required to have the above three attributes in the PSF.
- The tag appears in SD-UX GUI windows as basic name of software object.
 - The tag for a fileset cannot exceed 14 characters.
 - For bundles, the tag and title are from Corporate Price List (CPL).
- See this module's "Additional Notes" section for examples of PSF specifications for various objects.
- Refer to *Managing HP-UX Software*, Ch. 9, "Choosing Products to Package" for more details about, examples of and information about PSF specifications.
- Additionally, refer to swpackage(4) man page for a complete description of and example of PSF specifications.

Vendor specification example

```
vendor
  tag           Whatsit
  title         Whatsit Company
  description   </src/gizmo/info/vendor_description
end
```

Notes

- This page and the next few will provide the PSF specifications for an imaginary product, the Gizmo product, by the Whatsit Company.
- Because only HP can create bundles, the bundle specification example will be left out of this set of examples.

Product specification example

```
product
  tag          Gizmo
  title        Gizmo with widgets
  description   </src/gizmo/info/product_description
  copyright     </src/gizmo/info/copyright
  readme       </src/gizmo/info/readme
  revision      A.1.00.12
  architecture  HP-UX_B.10.00_700
  machine_type  9000/7??
  os_name       HP-UX
  os_release    ?.10.*
  os_version    *
  directory     /opt/gizmo
  is_locatable  FALSE
```

Notes

- The `copyright` and `readme` attributes specify pathnames to files that will be displayed when the SD-UX user requests this type of information about the product.
- The `revision` specifies the revision, or release, of the product.
- The `architecture` attribute, along with `machine_type`, `os_name`, `os_release` and `os_version`, are used to determine compatibility.
- The `directory` indicates the default, absolute installation location of this product on HP-UX.
- The `is_locatable` attribute is set to `TRUE` if the product can be installed into an alternate product directory. If set to `FALSE`, the product must be installed in the `directory` location.
- The keyword `end` will come after all of the subproducts and filesets for the products have been defined.

Subproduct specification example

```
subproduct
  tag           Runtime
  title         Runtime Configuration
  description    </src/gizmo/info/subproduct_description
  contents      GIZMO-MIX  GIZMO-AUX
end
```

Notes

- The `contents` attribute defines a list of filesets contained within the subproduct.
 - The fileset names should be delimited by white spaces.

Fileset specification example

```
fileset
  tag          GIZMO-HELP
  title        On-line Help for Gizmo
  description   </src/gizmo/info/fileset_help_desc
  revision     A.1.00.221
  is_kernel    FALSE
  is_reboot    FALSE
  prerequisite  gizmo.fset,r>V122,a=HP-UX,v=Whatsit
  corequisite  fubar.fset,r>V121,a=HP-UX,v=Whatsit
  configure    /src/gizmo/script/help.configure
  unconfigure  /src/gizmo/script/help.unconfigure
  directory    /src/gizmo/help=/opt/gadgets/bin
  file         *
end
```

Notes

- Attributes not previously described are listed below (all are optional).

Attribute	Description
is_kernel	TRUE if fileset contains OS kernel files, those used to build kernel
is_reboot	TRUE if fileset requires a system reboot after installation
prerequisite	Filesets which must be installed (configured) before this is installed (configured). Can be in dependent fileset's product or another product. Specify each prerequisite fully, using r, a, v.
corequisite	Filesets that must be configured before dependent fileset can <u>operate</u> correctly. Can be installed/configured after dependent fileset. Can be in dependent fileset's product or another product. Specify each using r, a, v.

- The configure and unconfigure attributes are discussed later.
- The directory and file attributes are discussed on the next page.

File and directory specifications

- In fileset specification, specify files and directories:
 - Recursively
 - directory /src/gizmo/bin
 - file *
 - Explicitly
 - file /src/gizmo/gizmo.exe
 - Relatively
 - file bin/gizmo.exe
 - file bin/gizmowidget.exe
 - By mapping source directory to target directory
 - directory /src/gizmo/bin = /opt/gizmo/bin
 - file *

Notes

- The directory attribute is set to the source directory in which subsequently listed files are located.
 - Use relative or absolute pathnames.
 - If use dir1 = dir2, then dir1 files copy into dir2. This makes dir2 the destination directory.
- The file attribute is set to the files to be packaged in fileset.
 - If used with directory, use file pathnames that are relative to directory.
 - Specify each file on a separate line.

Permissions specifications

- By default, destination inherits permissions of source
- Use `file_permissions` keyword to set explicitly
- Example:
 - To assign specified permissions to destination directory and all files copied into it:

```
file_permissions -m 755 -o root -g users
directory       /src/gizmo/bin=/opt/gizmo/bin
file            *
```

Notes

- There is no `file` attribute at product level. To assign specific permissions, specify them in each fileset of product.

Control script attributes

- Specify control scripts for product or fileset by:
 - using name of type of control script as attribute
 - using the pathname to control script as value
- Examples:

```
checkinstall /src/gizmo/script/checkinstall
postinstall /src/gizmo/script/postinstall
```
- Can also specify subscripts and data files associated with control scripts

Notes

- Additional control scripts or data files can be included with product or fileset, and stored with “real” control scripts.
 - Subscript called by supported control scripts
 - Data file read by control scripts
- Specify additional scripts using syntax:

```
path [=tag]
```

 - If tag not specified, swpackage uses `basename(1)` of source pathname as tag

Control script "DON'T's"

- Shut down or reboot system
- Initiate a kernel build
- Change system's init state
- Remove any control scripts or fileset's files
- Alter other fileset's files
- Modify startup/shutdown scripts or links
- Require user interaction
- Do nothing (scripts must have a purpose)
- Other prohibitions as specified in
Managing HP-UX Software, Appendix
B, "Control Scripts"



Notes

- Control scripts should alter startup/shutdown configuration variable scripts' values with extreme caution.
- Users' modifications should be overridden with extreme caution.
- SD-UX has several environment variables which are passed to the control scripts. These cannot be set by the scripts, but can be tested and used to set other variables.
 - See this module's "Additional Notes" section for a list of these.
- Direct stderr and stdout to STDOUT and STDERR, respectively not to console.
 - Includes output of commands called by control scripts.

Control script "DO's"

- Posix.2 (/sbin/sh) shell script
 - Use generic Posix.2-compatible features,
- Self-reliant, as install order not guaranteed unless there are prerequisites
- Use /usr/sbin/sw/control_utils functions
- Must be re-executable
- Track movement of any files it manipulates, for location by swremove
- Test control scripts extensively
- Other requirements as specified in *Managing HP-UX Software*, Appendix B, "Control Scripts"



Notes

- Korn shell script syntax will be supported.
- C-shell scripts will not be supported.
- Prerequisite fileset functionality is consistently available only to current fileset's configure script during installation, and to current fileset's unconfigure script during removal.
- The control_utils file comes with SD-UX and its general purpose utility functions are intended for use by any control script.
- Developer must include safeguards against failure for an invocation of a script that has already been executed.
- Use the IPD_addfile and IPD_delfile utilities in control_utils to assist swremove in locating and removing any files created by a fileset's control scripts when it removes the fileset.
- The burden of testing control scripts falls on the developer. A failed control script may cause the SD-UX command to fail.

Example configure script

```
# MAIN
UTILS=/usr/sbin/sw/control_utils
if [ -f $UTILS ]
then
    . $UTILS
else
    echo "ERROR:  Cannot find $UTILS"
    exit 1
fi
exitval=$SUCCESS

cond_mkdir 555 bin bin /var/opt/myapp
cond_mkdir 555 bin bin /etc/opt/myapp
mod_pathfile -a P /opt/myapp/bin
exit $exitval
```

Notes

Phases of packaging operation

- When the product's files, PSF and control scripts are available, run swpackage, which performs:
 1. Package selection phase
 - Reads PSF file
 2. Analysis phase
 - Checks for unresolved dependencies
 - Checks user's authorization to package
 - Checks to see if product is being repackaged
 - Analyzes disk space or other media size
 3. Packaging phase
 - Creates depot if necessary
 - Copies products into depot

Notes

- Reads PSF file to determine package's products, subproducts and filesets.
- Compares size of package with size of media to verify that package can be installed.
- Verifies that all dependencies are part of the package.
- Checks that user has correct permissions, as specified in ACLs, to perform packaging.
- Packages source files and information into a product object, and then creates/inserts product into distribution depot.
- If target is a tape media, it will create a temporary depot.
- When packaging to a distribution tape, it actually copies the depot contents to the tape.
- Can create a package using products in specified existing directory depot as the files to "re-package" for copying onto distribution tape, for example
- The distribution directory in which the package is inserted can be made into a registered depot by running swreg. Unlike swcopy, swpackage does not automatically register a depot that it creates.

Using swpackage command

- Uses syntax and options consistent with other SD-UX commands:

```
swpackage [-p] [-v[v]] [-V]
[-s product_specification_file|directory]
[-d directory|device] [-x option=value]
[-X option_file] [-f file]
[-Ssession_file]
[software_selections]
```

- Example:

- To package only COBOL product files listed in PSF file psf_file onto tape at /dev/rmt/0m:

```
swpackage -s psf_file -d /dev/rmt/0m
-x target_type=tape COBOL
```

Notes

- Follow -s with either:
 - name of PSF describing intended package, or
 - pathname of existing depot directory, to package contents
- Follow -d with intended location of package, either:
 - distribution directory pathname, or
 - actual device name of tape drive, for example.
- Use software_selections to specify (re)packaging only those software selections from full set of software defined in PSF.
- Use -vv to output “very verbose” information as command is running.
- Use -p to preview packaging process, to determine problems first.
- To package products by inserting references to original source files, instead of copying directly into target depot, set defaults option swpackage.package_in_place to true.
 - Useful for development or test environment.
- Refer to sd(5) man page for a complete list of defaults options.
- For permissions (ACL entries) to package, see “Additional Notes”.

Modifying an existing
product: `swmodify`



Modifying product: `swmodify`

- Modifies existing products in target root or depot
- Provides direct interface with a depot's catalog or a root's Installed Products Database (IPD)

Option	Description
<code>-d</code>	modifies depot, using <i>target_selection</i>
<code>-s psf_file</code>	modifies specified <i>psf_file</i>
<code>-u</code>	deletes specified <i>software_selections</i> or specified attributes
<code>-a attribute[=value]</code>	provides attributes/values to modify, delete, add

Notes

- Alters information about files located in the specified PSF, from the depot catalog for the specified depot or from the IPD for the specified root.
- Does not alter the actual files making up the product or fileset.
 - Adds new or deletes existing objects of product or fileset
 - Modifies attribute values of specified *software_selections*
- If PSF specified with *software_selections*, modifies those only.
- If PSF specified with no *software_selections*, modifies all in PSF.
- If no PSF, uses *software_selections* in *target_selections*.
- Use `-u` with no `-a attributes`, to delete *software_selection* definitions from *target_selection's* catalog or IPD.
- Use `-u` with `-a attributes`, to delete attribute (or specified values).
 - If no `-u`, then `-a attributes` will be added, or values modified.
- Use `-p` to preview without modifying.
- Uses standard SD-UX command options, in addition to those above.
- For examples of using `swmodify`, see "Additional Notes" section.
- Most often, will probably be used at "configuration" time.

Converting fpkg to swpackage

- Use fpkg2swpkg to translate fpkg network media or PSF to swpackage PSF with these steps
 1. Execute fpkg2swpkg command.
 2. Modify and/or develop control scripts for SD-UX.
 3. Edit the new swpackage PSF to alter defaults and add information not provided in fpkg psf.
- Syntax of fpkg2swpkg:

```
fpkg2swpkg [-m mediatype] [-d dest] fpkg_pathname
```

mediatype is source: psf (fpkg PSF) or network (fpkg netdist server)
dest is path of new swpackage PSF
fpkg_pathname indicates path to source

 - Runs on both HP-UX 9.0x and 10.0

Notes

- Default media type is psf.
- If the source is a netdist server, then fpkg2swpkg first calls fpkg -r to create an fpkg PSF from it.
 - The fpkg utility must exist on system for this to occur
- Default destination is ./swpackage.psf.
- The fpkg_pathname can be relative or absolute for fpkg PSF filename and must be absolute for fpkg netdist directory.
 - This overrides -m mediatype
- Several fpkg keywords are not used by SD-UX and will be ignored.
 - A warning will occur during fpkg2swpkg process to indicate this
- Several swpackage keywords have default settings which the user may wish to edit after fpkg2swpkg process has created new psf.
- Most fpkg customize and decustomize scripts will need to be edited after this process to work within an SD-UX environment.
- Refer to fpkg2swpkg(1) man page for a detailed description of how to use this utility and the potential problems with conversion.

Tips and tricks

- In configure control script:
 - Use `mod_pathfile`, a `control_utils` function, to add newly installed product's command path to system path
- Use PSF file keyword options to set file permissions, such as owner, group, mode for individual files
- While `swinstall` will create `/opt/app_dir` directory, `swremove` will not remove it
 - Use `postremove` script for product to do this, unless directory is also used by other products



Notes

- If the empty directory was only used by one fileset, then use `postremove` script for fileset to remove it.
- In actuality, most developers are not using any product control scripts, just fileset control scripts.

Summary and review

- The swpackage command uses PSF or contents of specified depot directory to create package which is placed in specified distribution directory
 - Distribution directory can be registered, using swreg, as depot
- The PSF describes the attributes of the product and its components in detail
- Control scripts can be developed to customize the software management of the product
- Packages can be re-packaged to distribution tape
- Use swmodify to alter the IPD information for a package

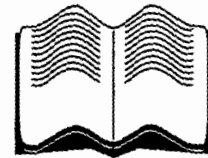
Notes

For more information



For more information

- *Managing HP-UX Software*
- The man pages for sd(4), sd(5)
- The man pages for swpackage(4) and swpackage(8), swmodify(8)



Notes

You will package a product by creating its PSF and using swpackage. Then, you will register the newly created depot, and install and test the new product.

1. Change to the directory:
 /opt/deltaclass/labs/echoapp.
 Source files for the new product EchoApp are here.
 When properly installed, Echoapp will be in:
 /opt/echoapp
2. List the subdirectories in this directory.

Review this table of contents for the directories:

Subdirectory	Contents
PSF	template PSF
commands	executables for EchoApp
doc	online documentation for EchoApp
script	control scripts for EchoApp
info	description files, etc. for EchoApp's PSF

3. Change to info directory.
View all *_desc files for information about product structure, including names of files in each fileset.
Which file contains the names of all filesets?

Which file contains information about the vendor?

Who is the vendor?

4. Complete this table, based on information in files:

Fileset name	Files in fileset

5. Change to PSF directory.

Enter: `cp template.psf echoapp.psf.`

6. Modify `echoapp.psf`, referring to the example PSF information provided in any of these locations:

The `swpackage(4)` man page

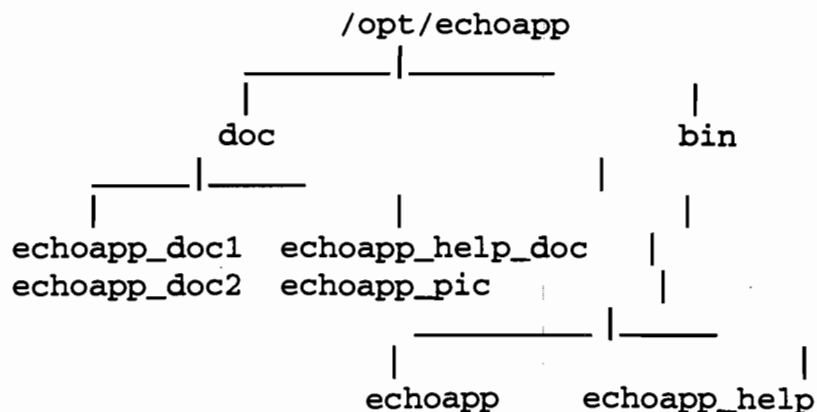
Managing HP-UX Software, Ch. 9

This module's "Additional Notes" section

NOTE: You will save yourself a lot of time by carefully typing in all filenames, and attribute values.

NOTE: Use explicit file permissions.

NOTE: The `/opt/echoapp` directory should look like this, when installed successfully:



-
7. When you are done, run `swpackage` in preview mode:

```
swpackage -pvv -s echoapp.psf -d /tmp/echoapp
```

You may want to redirect output to a file.

What does the `-d` option mean?

What does `-vv` do?

8. After correcting any errors or warnings, run `swpackage` in preview mode again, until there are no errors or warnings.
9. Run `swpackage` for real:

```
swpackage -vv -s echoapp.psf -d /tmp/echoapp
```
10. Register the depot per instructions.
11. Run `swinstall` to install EchoApp, remembering to select the new depot as the source.
12. If it installed correctly, you should have several new directories, including:

```
/opt/echoapp  
/var/opt/echoapp  
/etc/opt/echoapp
```
13. Re-login. The new path `/opt/echoapp/bin` should be added to the system path.
How can you verify the new path is there?
-
14. Test out the new application, by entering:
`echoapp_help`

What happens? _____

Enter: echoapp _____

15. Extra credit:

View the configure and unconfigure scripts.

What do these do?

How did the new path /opt/echoapp/bin get added to the system path?

How did the new /var and /etc directories get created?

16. Extra credit:

Look through the new depot: /tmp/echoapp and the new addition to the IPD:

/var/adm/sw/products/EchoApp

What are the differences between the files contained in the new depot and the new IPD addition?

17. Extra credit:

Read the man page and this module's page on swmodify, and try it out on your new product's PSF.

Change the vendor name to your name.

What command do you enter to do this?

This page intentionally left blank.

Additional notes for “Software Distributor Packaging” module

Environment variables

Environment variable	Defines
SW_PATH	path to look for executables
SW_ROOT_DIRECTORY	root in which session is operating
SW_LOCATION	location of the product
SW_CONTROL_DIRECTORY	directory containing script
SW_INITIAL_INSTALL	cold install in progress
SW_DEFERRED_KERNBLD	software installed to non-/ directory
SW_KERNEL_PATH	path to kernel
SW_SYSTEM_FILE_PATH	path to kernel's system file
SW_SOFTWARE_SPEC	full specification of current software
SW_SESSION_OPTIONS	path to file listing filesets selected

Notes

- All control scripts will be passed these environment variables.
- Scripts should never SET these, just test and use them to create other environment variables.
- When prefixing path variables, do not use a "/" as the leading character in the path, as this will expand to "/" which can cause a failure in string comparisons.
 - All path variables exported to scripts will have a trailing "/"

Examples of using swmodify

- To add files to an existing fileset:

```
swmodify -x files='/tmp/file1 /tmp/file2' PROD.FSET
```

- To delete fileset definition:

```
swmodify -u PROD.FSET
```



Notes

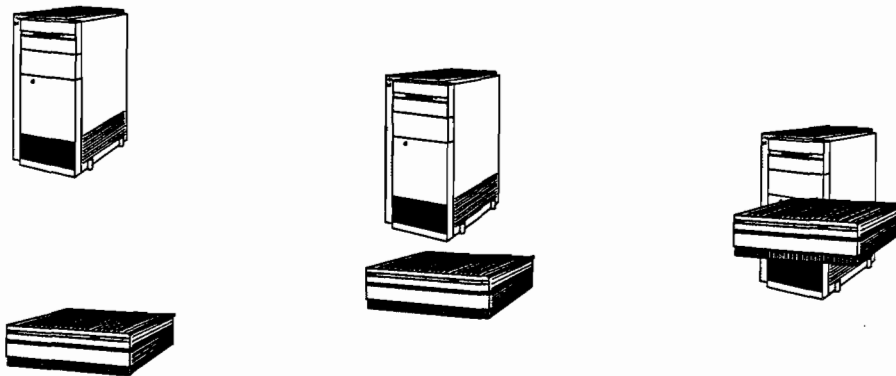
ACL permissions to package

- Unless superuser is packaging, these ACL permissions are necessary to package:

Permission	Object	Description
insert	local host	to install new depot
insert	target depot	to add product to existing depot
write	product	to modify existing product
write	depot	to change existing target depot attributes, if PSF specifies this

Notes

S700 and S800 I/O Convergence



Module goals:

The purposes of this module are to describe I/O convergence for the S700 and S800, and to explain how to administer I/O devices on HP-UX 10.0.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the motivations for I/O convergence.
2. List and describe the characteristics of I/O convergence at HP-UX 10.0, including what has been and what has not been converged.
3. Describe the components of I/O convergence, such as General I/O System (GIO) and Context-Dependent I/O Systems (CDIO).
4. Explain how device special file names have changed.
5. Administer I/O devices with `ioinit`, `ioscan` and the `*sf` utilities.

Motivations for convergence

- **Provide software infrastructure for running both S700 and S800 I/O on HP's converged hardware**
- **Move S800 driver environment toward industry norm**
 - **New drivers to be written in S700 style**
- **Reduce overhead of maintaining two HP-UX labs to create, support and maintain drivers**
- **Increase sales by reducing customer confusion**
 - **Competitors have used the differences between S700 and S800 to take away sales from HP**

Notes

- **We are moving to industry-standard driver architecture by making S700-style I/O the HP-UX driver environment of choice.**
 - **S800-style drivers are more similar to MPE I/O (the 3000 series) than industry-standard Unix.**
 - **S700-style drivers are based on the Motorola architecture and are more like industry-standard Unix.**
- **The converged I/O provides more flexibility for HP to move quickly in the direction of new technology.**
 - **While we do not know exactly what standards will take hold in the future, having converged I/O will speed up development of drivers.**
- **The new boxes allow co-existence of the S700 and S800 I/O systems. These new systems, the J and K Series, will be available 1Q95.**

Features and benefits

Features

- For both S700 and S800, there is now a single:
 - Set of access methods across both architectures
 - Set of kernel configuration utilities
 - Convention for creating and naming device files
 - Scheme for disk layout
 - Set of high availability features
- Additionally, it supports:
 - Coherent I/O services on new systems

Benefits

- Write only one driver, not two
- Easier-to-use system administration tools and documentation

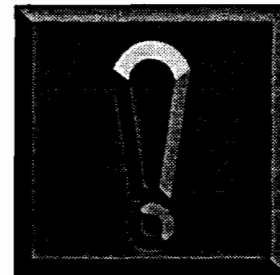


Notes

- Both driver architectures – WSIO (S700) and SIO (S800) – architectures are now in a single HP-UX kernel.
- A number features (Logical Volume Manager, mirroring, and diagnostics) previously available only on one platform have been made available on both, with very little effort.
- HP's S700 J Series (Skyhawks) and S800 K Series (Kittyhawks) provide coherent I/O hardware. Driver software can take advantage of this for improved performance.
- System administrators will (eventually) no longer have to deal with two sets of documentation and utilities.

Overview of impacts

- Most drivers will require minor modifications
 - Exception is S700 EISA, non-SCSI
 - New H/W may implement I/O Coherence
- Major and minor numbers have changed
 - Many are allocated dynamically
- Performance impact is minimal
 - Kernel <-> HW mapping done at boot-up

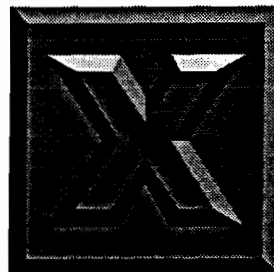


Notes

- We will continue to leverage off the huge existing investment in drivers.
 - Drivers were not rewritten except where required to maintain functionality. S800 drivers and S700 HP-internal drivers were modified.
 - S700 EISA, non-SCSI drivers are not required to be modified — although HP still recommends that they be modified. All other drivers on S700 and S800 are required to be modified to conform to the new architecture.
 - The *HP-UX Driver Development Guide* tells much more about this.
- Changes to major and minor numbers are covered later in this module.
- Coherent I/O services are covered later in this module.

I/O features *not* in HP-UX 10.0

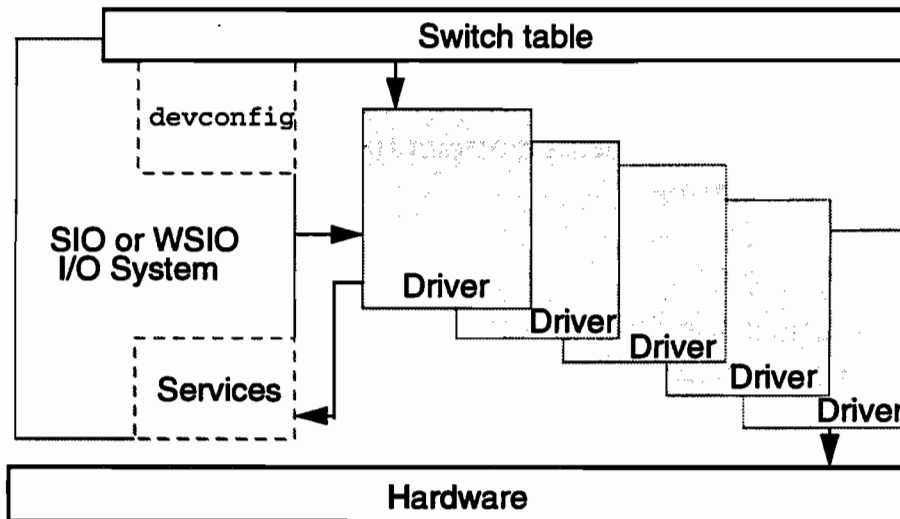
- Identical behavior on S700 and S800
 - 9.0x differences still exist
- Merged S700 and S800 SCSI drivers
- Merged S700 and S800 peripheral lists
- Merging of other peripheral types
 - Centronics
 - HP-IB
- Open I/O
- Support for dynamic loading and unloading of drivers



Notes

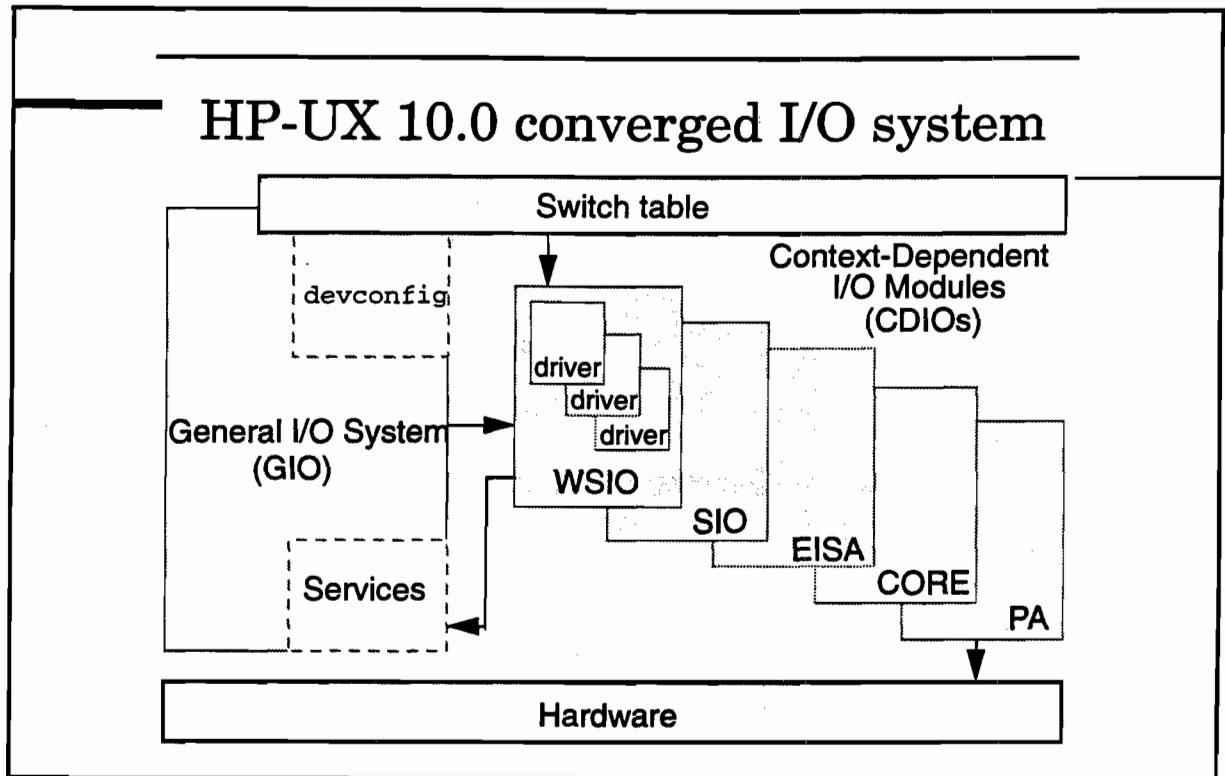
- All of these features are not available in HP-UX 10.0, but the framework is in place to provide support for them in future HP-UX releases.
- There is not yet one driver for everything, but eventually there will be one driver per function.
 - Need to leverage from existing drivers
- Currently, only the framework is in place for Open I/O.
 - *HP-UX Driver Development Guide* is being revised heavily for HP-UX 10.0, to explain how to get drivers to fit into the new framework
 - We are investigating a full Open I/O program
- Drivers are not yet able to install themselves. They need to be added manually to the system file used to rebuild the kernel.

I/O Architecture in HP-UX 9.0x



Notes

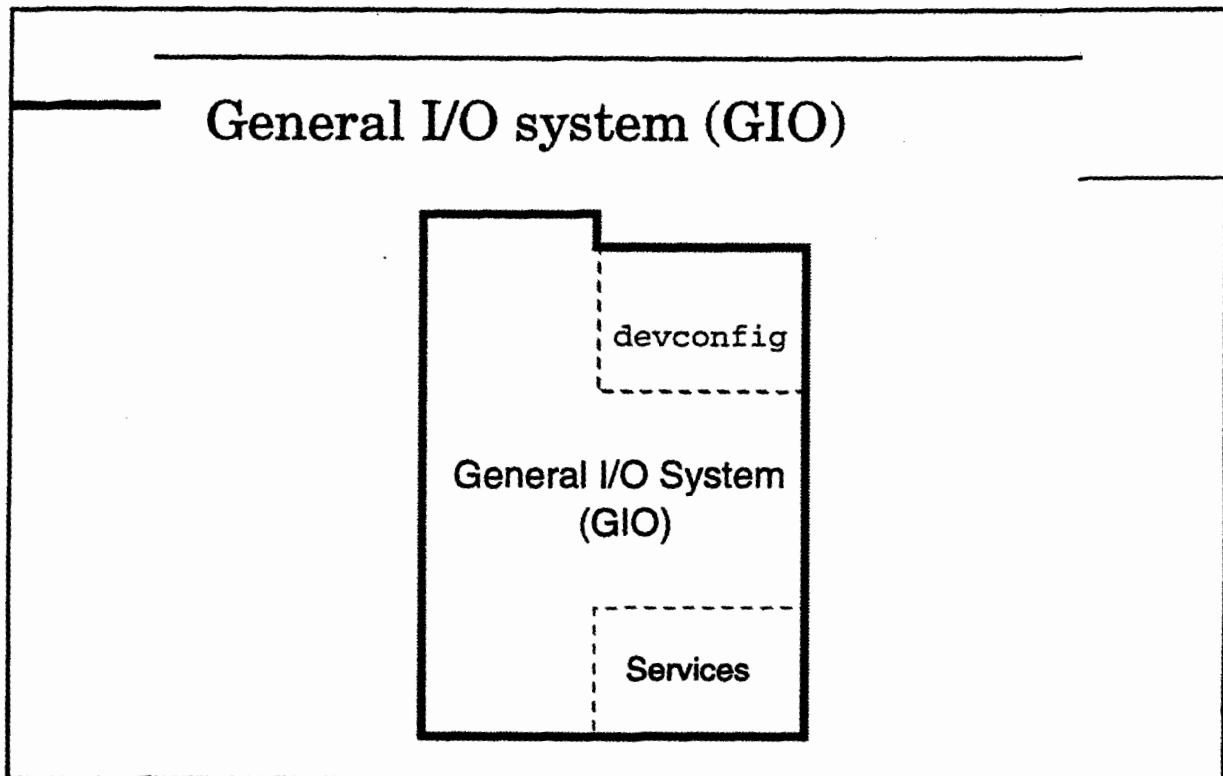
- The architecture of the I/O systems on S700 and S800 versions of HP-UX prior to 10.0 looked the same, but each was supported only on its particular platform.
- The SIO and WSIO schemes used very different mechanisms for implementing their drivers
 - The SIO scheme used a stack-based driver architecture and message-passing between the driver layers.
 - The WSIO architecture used two-layer (user & interrupt) model with a set of well-defined entry points to its drivers.
- Note that in both, once the I/O tree was built, access to the drivers occurred directly from the switch tables.



Notes

- Both the SIO and WSIO architectures now coexist: hardware-specific or driver environment-specific functionality is hidden from rest of system behind well-defined interfaces.
- The I/O system is broken into two main sections, the General I/O System (GIO) and one or more Context-Dependent I/O modules (CDIOs).
- The GIO is responsible for all functionality which is global to system and provides primitive services for CDIOs.
- CDIOs contain all bus-specific and/or driver environment-specific functionality.
 - Configured into a system only as necessary

Note: While both architectures are supported, it is expected that new drivers will be written in the S700 style, since it's closer to the norm for UNIX drivers. The driver writers guides have been merged at 10.0, and only discuss this style.

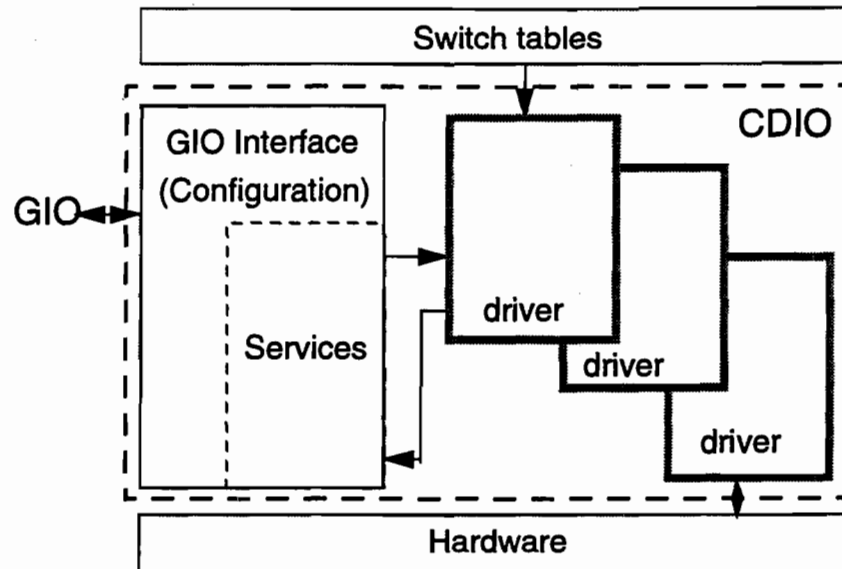


Notes

- The GIO:
 - Manages global I/O resources and data structures:
 - I/O tree nodes
 - Block and character switch tables
 - Kernel device table (KDT)
 - Drives system configuration process
 - Provides an interface to system administration utilities
 - Uses devconfig pseudo driver
 - Provides primitive global services:
 - Inter-CDIO communication
- The GIO does not directly interact with device drivers or devices.

Note: It may eventually provide dynamic driver loading and unloading, but this is not a 10.0 feature.

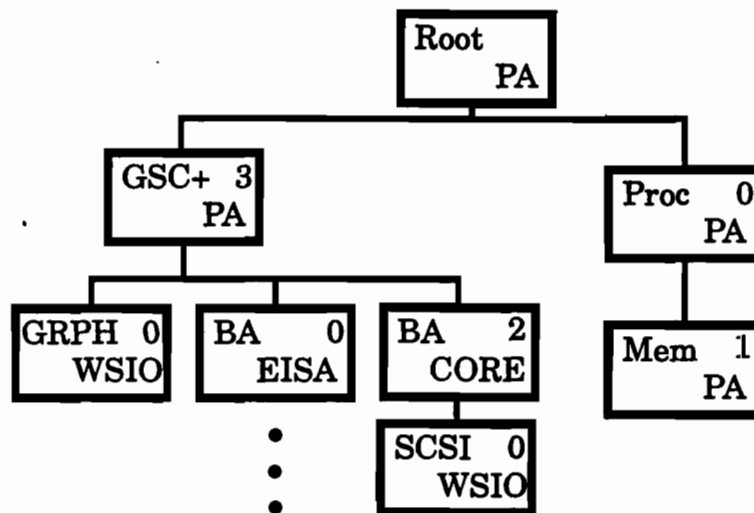
Context Dependent I/O Systems



Notes

- CDIOs come in two flavors:
 - *Bus-nexus*, providing bus-dependent services to other CDIOs
 - PA, CORE, EISA
 - *Driver-environment*, providing drivers within a defined environment
 - SIO, WSIO
- Each CDIO has up to five basic components: GIO interface, Inter-CDIO communication interface, Driver services, Drivers (appearing only in driver-environment CDIOs), I/O resource management
- CDIOs are configured into the I/O tree of a system only as necessary to support the H/W on that system
 - A S700 system would contain WSIO, CORE and PA CDIOs; it may contain EISA, but not SIO
 - A Nova (S800) would contain SIO and PA CDIOs, but no WSIO or CORE

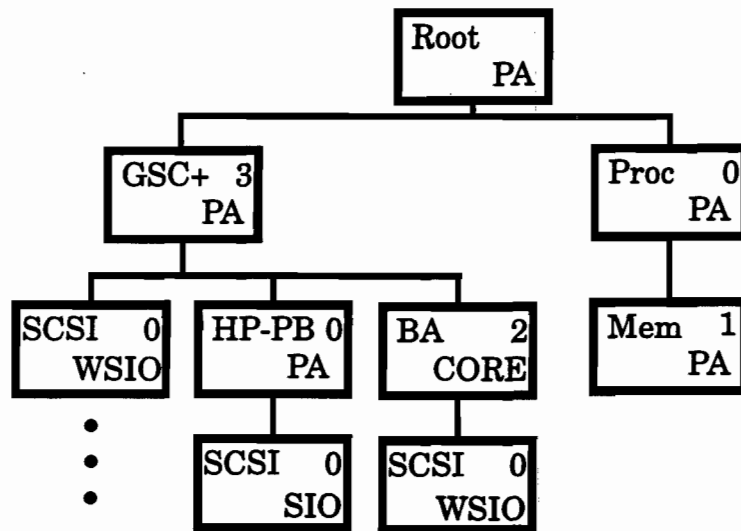
Example – S700 I/O Tree



Notes

- This is *part* of the I/O tree for a Series 700 system running HP-UX 10.0. It's not intended to be strictly correct, but it does illustrate the way that software modules are configured.
- Note the numbers of the nodes, which represent the instance numbers. Instance numbers are roughly analogous to LUs on the S800 machines prior to HP-UX 10.0.
- Instance numbers for new modules are assigned at system boot, and remain consistent across reboots. The mappings between the numbers and their corresponding modules is stored in the `/etc/ioconfig` file. This file is initially built when the system first boots HP-UX 10.0, and gets added to if a system's hardware configuration changes. Note that if its contents are not lost, instance numbers will not change.

Example – K-Series I/O Tree



Notes

- Here's a portion of the I/O tree for a K-Series machine. Note the simultaneous existence of SIO and WSIO modules in the tree.

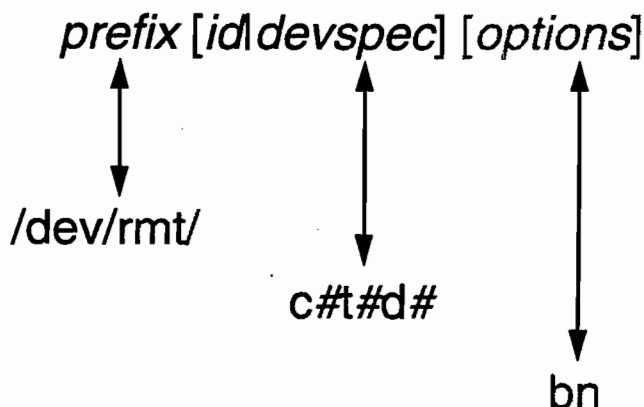
Admin issues – special file changes

- Single device file name convention
 - Follows USL SVR4 conventions
 - Names incorporate physical location info
- Converged major & minor numbers
- Most major numbers changed
 - Too many drivers, lots of S700/S800 conflicts
 - Many are assigned dynamically at 1st boot
- Most minor numbers changed
 - No LU (S800) in converged framework
 - Owned by drivers; not interpreted by I/O system

Notes

- There are no industry standards for device file names, but there are conventions. The convention we follow is the same used by SunOS 5.x.
- Major and minor numbers are in the *Configuring HP-UX for Peripherals* manual.
- There are not enough possible major numbers (256 character, 256 block) to re-assign all conflicting S700 and S800 major numbers, in addition to providing new major numbers.
 - Many major numbers will be dynamically allocated during I/O system configuration.
- New policy for minor numbers: In general, LUs map to card instances.
 - The device's minor number is owned by the driver instead of being interpreted by the I/O system. This removes the dependency between drivers and the converged I/O system
 - This does not prohibit SIO and WSIO CDIOs from providing services that interpret the old minor number format

Device file name convention

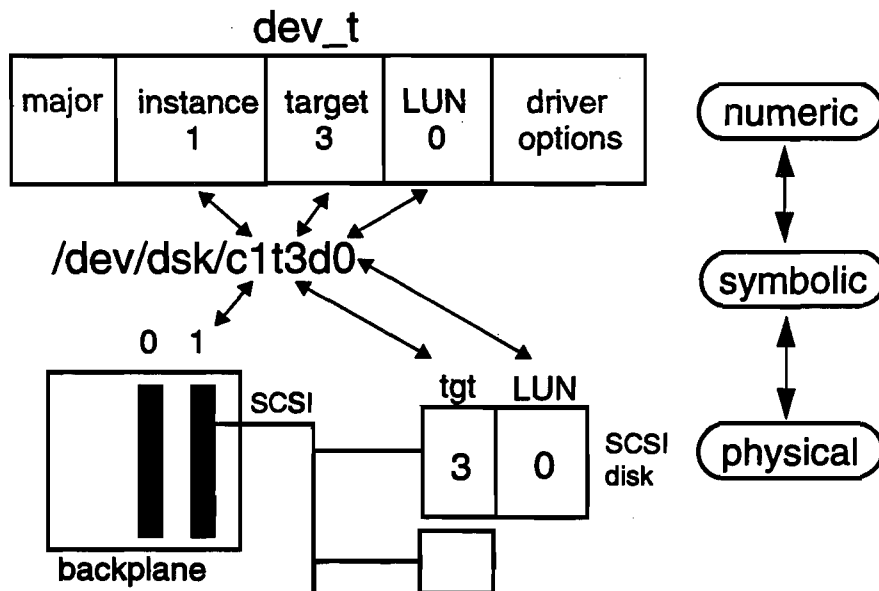


Notes

- Device files are still in the */dev* directory. The *prefix* may be a directory relative to */dev*, such as *pty* or *rmt*. It should indicate the device type. If *prefix* is insufficient, use either *devspec* (preferred) or *id* but not both.
- The *id* is a logical number or designator assigned by the system administrator. It has no relationship to the card number, bus addresses or previous S800 LUs. The *id* is available for pseudo drivers or certain system default device files like */dev/rmt/0m*.
- The *devspec* indicates hardware path information:

<i>c#</i>	Interface card instance (the 'c' is optional if it is the only address element)
<i>t#</i>	Target address on remote bus (for example, SCSI target, HP-IB address)
<i>d#</i>	Device unit at target (for example, SCSI LUN, HP-IB unit)
- Options are device-specific — for example, *h* for high density tape, *_lp* for a printer, or *s#* for a disk section number.

Example – Device Mapping



Notes

- The `c#t#d#` convention comes from USL's SVR4.
 - The `c#` is assigned by the I/O system; `ioscan` reveals it
 - May be from one to three digits
 - At boot, the system scans the backplane and attaches a card instance number to that device
 - Keeps the same value across reboots (thanks to `/etc/ioconfig`)
 - Has no direct correlation with path to physical card; not a physical hardware address — think of it as a randomly-assigned value (it's really a node number in the `iotree` kernel data structure)
 - The `t#` is base 10; can be one or two digits; typically is in range 0–15
- The `options` convention is not USL, but is consistent with previous S700 and S800 systems and with SunOS 5.x.

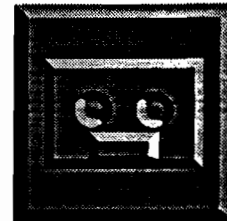
Example – Tape device naming

System	Naming convention
9.0x S800	/dev/rmt/<lu><options>
9.0x S700	/dev/rmt/[c#d]#<options>
10.0 converged	/dev/rmt/c#t#d#<options>

- **Examples:**

- /dev/rmt/c1t3d0BEST
 - /dev/rmt/c1t3d0sQIC150nb
 - /dev/rmt/c1t3d0D8MM8500C

- Use `mksf`, not `mknod`, to create tape device files



Notes

- The tape device file name convention reflects any and all configuration options available in the minor number. Minor numbers and major numbers were radically changed at 10.0. Configuration options previously offered through only the S700 or S800 drivers are incorporated into one common set of options.
 - See `mksf(1M)` for description of options, including exceptions to the above statement
- Short file name systems will need to use links.
- Default density (BEST) includes compression on devices which support it.
- The S700 diagnostic path (/dev/diag/mt) was selected instead of a V.4-style path name.
 - This preserves compatibility with 9.0x names by linking to the corresponding device instance

Recognized tape devices

Tape device	S700	S800
7980S, 7980SX	Yes	Yes
Quickstep	Yes	Yes
Macbeth	Yes	Yes
Tardis	Yes	Yes
Othello	Yes	Yes
Archive Viper	Yes	
Exabyte 8505		Yes
Wangtek 5525		Yes
Wangtek 51000		Yes
Fujitsu		Yes
Storagetek 4280		Yes

Notes

- S800 and S700 behaviors are consistent with 9.0x
 - The S800 tape driver is designed to be "closed." It will open only those tape devices that are on its list of recognized devices.
 - The S700 tape driver is designed to be "open." It will attempt to open any tape device, whether or not it is a recognized device mode. Therefore, the S700 might work with tape devices other than those listed above, but the S800 will work with only those listed.
- Additionally, the Storagetek 4781 and 3480 are *referenced*, not fully supported by the system. In other words, it is supported to a limited degree; a third-party application is needed to read/write tapes in that format.
- More information on tape devices is in the *Configuring HP-UX for Peripherals* manual.
- The configuration process for tape drives changed significantly. The `mtinit` program (run at boot from `/etc/inittab`) now does tape device set-up for the kernel at each boot.

Example – Disk device naming

System	Naming convention
9.0x S800	/dev/[r]dsk/c<lu>d<unit>s<section>
9.0x S700	/dev/[r]dsk/[r]c#d#[l#][s#]
10.0 converged	/dev/[r]dsk/c#t#d#[s#]

- Discourage use of sections (s#)
- Examples:

/dev/dsk/c0t6d0
/dev/rdsk/c3t2d4



Notes

- This convention is most similar to Sun's style.
- Do not use s# unless it is absolutely necessary for backwards compatibility. Use of section numbers is supported only for disks that were partitioned on 9.0x (or are replacing identical disks partitioned on 9.0x).
 - The section number can be 0–15.
 - Section 0 refers to the entire disk under 10.0.
 - Use insf(1m) to create device files for disk partitions.

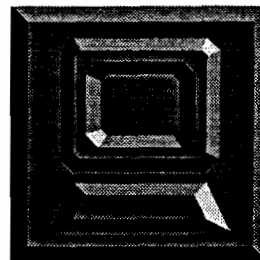
Example – Serial device naming

System	Naming convention
9.0x S800	/dev/tty<lu>p<port>
9.0x S700	/dev/tty<XX> (where XX is line identifier)
10.0 converged	/dev/tty<D>p<port> (D = card instance, p = port number)

- **Examples:**

/dev/tty12p3

/dev/tty0p0



Notes

- All other naming conventions for device files are described in detail in the manual *Configuring HP-UX for Peripherals*.

Overview – I/O utilities

- Commands, behavior are same on S700 and S800

`mk_kernel`

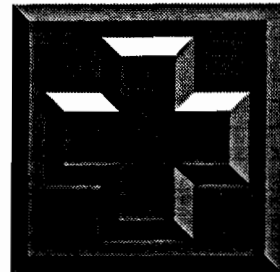
`config`

`ioinit`

`ioscan`

`insf, lssf, mksf, rmsf`

`mediainit`



Notes

- Most of these are covered in more detail on the next few pages.
- See the manual pages for further details on each command.

System generation convergence

9.0x	10.0 replacement	Purpose
uxgen, regen, config	mk_kernel	Builds both S700 and S800 kernels
/etc/master	/usr/conf/- master.d/*	Contains master files for each subsystem
S800, dfile	/stand/system	Text file of system drivers & parameters (common format based on 9.0x dfile)
/hp-ux	/stand/vmunix	Kernel executable

Notes

- The new `mk_kernel` script calls `config`, which reads all the files in the new master directory `/usr/conf/master.d` for device specifications, tunable parameters, and other kernel information.
 - **Changes to these master files are not supported.**
The files will be overwritten with each major update to the OS.
 - Driver writers should create their own master file pieces and deliver them to this directory as part of their products.
 - See the `master(4)` man page for a complete description of the new master files.
- More about `mk_kernel` is in the "Kernel Generation" module.

Converged ioinit features

- Now checks consistency of kernel I/O tree (derived from `/stand/ioconfig`) and `/etc/ioconfig` at boot (`-i`, `-r` options are used)
 - If consistent, invokes `insf` to create special files for new devices
 - If inconsistent, updates `/stand/ioconfig` and reboots system
- 9.0x `ioconfig` files converted to new format automatically at upgrade
 - *Upgrade not available at 10.0 release*
- Also may be used to force reassignment of instance numbers of existing entries (`-f` option)

Notes

- The GIO reads `/stand/ioconfig`, which holds I/O system information that must be preserved across reboots:
 - Mappings of instance numbers to hardware paths
 - I/O address space usage
- The master `ioconfig` file is kept in `/etc`, but it is copied to `/stand/ioconfig` to avoid access problems for NFS diskless clients. (Note that NFS diskless is not available at first 10.0 release.)
- Note that there's also a `-c` option, which will rebuild `/etc/ioconfig` from kernel data structures. This option may be used to restore `/etc/ioconfig` when the file is lost or corrupted and no backup copy is available.

Converged ioscan features

- Removed kernel I/O system dependency
 - Uses libIO for kernel information
- Common interface for S700 and S800
 - Adopted from S800
- Output usability and readability improved
 - New -F option lists more items separated by colons; useful with awk
- Use -I option to specify instance number at configuration
 - The -1 option is obsolete

Notes

Example (partial, from 9000/715):

```
# ioscan -f
```

Class	I	H/W Path	Driver	S/W State	H/W Type	Description
bc	0		root	CLAIMED	BUS_NEXUS	
graphics	0	0	graph3	CLAIMED	INTERFACE	Graphics
ba	0	2	bus_adapter	CLAIMED	BUS_NEXUS	Core I/O Adapter
ext_bus	0	2/0/1	c700	CLAIMED	INTERFACE	Built-in SCSI
target	0	2/0/1.6	tgt	CLAIMED	DEVICE	
disk	0	2/0/1.6.0	sdisk	CLAIMED	DEVICE	HP C2247
lan	0	2/0/2	lan2	CLAIMED	INTERFACE	Built-in LAN
hil	0	2/0/3	hil	CLAIMED	INTERFACE	Built-in HIL
tty	0	2/0/4	asio0	CLAIMED	INTERFACE	Built-in RS-232C
tty	1	2/0/5	asio0	CLAIMED	INTERFACE	Built-in RS-232C
ext_bus	1	2/0/6	CentIf	CLAIMED	INTERFACE	Built-in Parallel
audio	0	2/0/8	audio	CLAIMED	INTERFACE	
ba	1	4	eisa	CLAIMED	BUS_NEXUS	EISA Adapter

Converged *sf commands

- The preferred method for *expert* managing and displaying information about I/O devices:
 insf, mksf, rmsf, lssf
- Common interface for S700 and S800
- Default target is now /dev

New option	Purpose
-D <i>directory</i>	Overrides default of /dev
-q	Quiet, to reduce output during device file creation and deletion
-v	Verbose, to increase output of commands
-I <i>instance</i>	Replaces -l lu

Notes

- Encourage customers to use SAM to make special device files!
- In 9.0x, these commands manipulated device files in the current directory. Files will be created in /dev by default in HP-UX 10.0.
- Most core drivers do not support using rmsf to remove device files from the system (i.e. /etc/ioconfig); it will only remove the /dev special files.
 - Use -q to suppress the warning

Converged disk layout

Why?

- Hardware convergence
 - New systems like K Series support both NIO and CORE; layout and functions need merging

Summary of changes

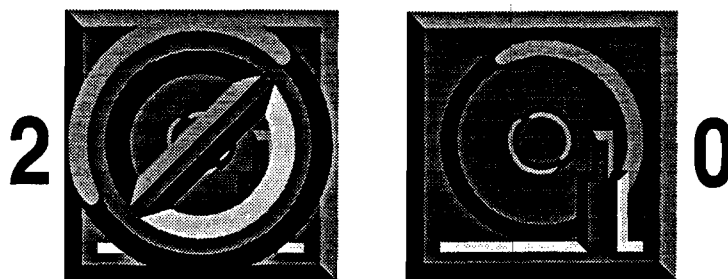
- LVM strongly recommended for all systems
- Upgrade multi-disk SDS arrays to LVM
- Definition of sections 2 and 0 swapped
- Backward compatibility for single-disk SDS and hard partitions provided by Compatibility Pseudo Driver (cpd)

Notes

- LVM and SDS topics are covered in more detail in the “Logical Volume Manager” module.

Hard partition convergence

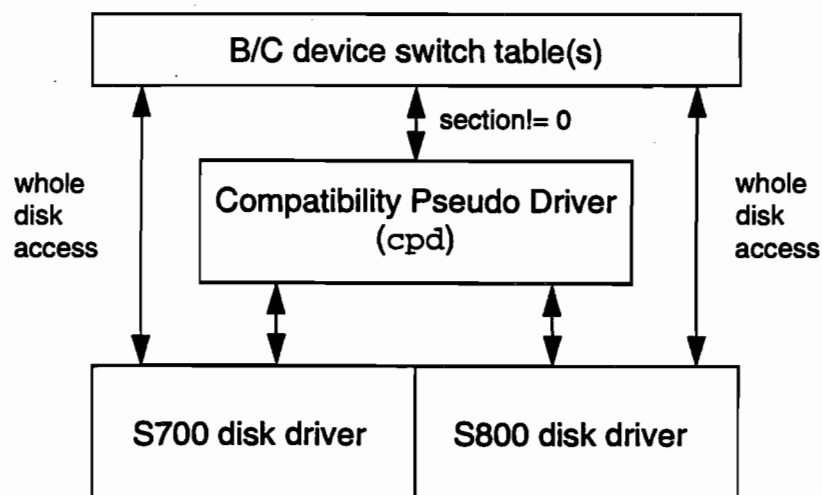
- Section 2 and section 0 are swapped:
Entire disk is not section 2, but 0
- Hard partitions on new drive models not supported



Notes

- Note that the use of partitions on disk drives is only supported to allow for pre-existing configurations. New systems and hardware only support Logical Volumes.
- LVM is covered in the “Logical Volume Manager” module.

Compatibility Pseudo Driver



Notes

- The cpd supports fixed-partition and single-disk SDS access.
 - Invoked automatically when the device file ends in "s#"
 - Note that multi-disk SDS configurations (the norm) *must* be converted to LVM. Only single-disk SDS is supported by the cpd.
- LVM uses whole-disk access. This means that cpd is not called.

Developers' issues – Major Number assignment

- Still represents slot in bdev/cdev switch table
- Most allocated dynamically
 - Applications generally shouldn't assume value
 - May be reserved for critical apps
- Defined in device tables
 - Located in /usr/conf/master.d/*
 - Value of "-1" in table indicates dynamic assignment

Notes

- Example table entries (from core-hpux)

* Driver	Block major	Char major	Required for minimal system
*			
asio0	-1	1	
cpd	-1	-1	1
kepd	-1	227	1
dev_config	-1	69	1
klog	-1	189	1
cn	-1	0	1
mm	-1	3	1
pty0	-1	16	1
pty1	-1	17	1
* Core drivers			
lasi	-1	-1	
asp	-1	-1	

New dev_t format

Table 1: Old S700 (WSIO)

Bits 0-7	8-11	12-15	16-19	20-31
major	SMB	slot	fn	drv_opts

Table 2: Old S800 (SIO)

Bits 0-7	8	9-15	16-19	20-31
major	T	flags	mgr_index	drv_opts

Table 3: New converged WSIO & SIO

Bits 0-7	8-15	16-19	20-23	24-31
major	instance	target	LUN	drv_opts

Notes

- The example above is SCSI. For HP-IB and HP-FL, use *address* instead of *target*, and use *unit* instead of *LUN*.

Major	One of 256 major numbers
Card instance	One of 256 possible card/function instances; note that this is the interface <i>card's</i> instance number, not the <i>device's</i>
Target	One of 16 SCSI targets
LUN	One of 16 SCSI LUNs
Dvr_opts	Optional bits for driver's use

- All cards that support remote devices (namely SCSI, HP-IB, FL) share the same class (pool of numbers). For example, all LAN cards are in a class.
- The instance number is the 9.0x S800 LU concept extended to cover interface cards and EISA. It reflects the backplane slot and priority.

Required changes to drivers

- Add driver header
- Create a driver_install routine for the driver
- Remove hard-code, bus-dependent calls
- Add isc_claim() call to driver_attach() routine (interface driver only)
- Adjust driver_init() routine to check for cons_mux_dev (if claiming console)
- Update minor number references
- Replace ISC-table accesses with get_isc() calls
- Add (*minor_build)() and (*minor_decode)() routines
- Add spinlocks and semaphores

Notes

- This is a summary of the steps, see the HP-UX Driver Development Guide for details.

Coherent I/O

- HW feature of J & K Series
 - I/O subsystem supports cache coherency in H/W
 - Will improve performance on MP systems
 - Drivers have to know about it, though
- Drivers must be modified to support coherent and non-coherent (legacy) systems
 - Modifications are straightforward
 - Smart services determine hardware type and behave appropriately, which allows drivers to run on either coherent or non-coherent hardware without recompilation

Notes

- “Coherent” means that the I/O subsystem has H/W that allows it to participate in H/W-based cache coherency mechanism of MP PA machines.
- In a coherent environment, drivers should not manage the cache explicitly (flush and purge as it sets up DMA, etc.). If a driver flushes and purges manually, it will adversely impact performance.
- At this point, most HP 9000 systems are not I/O Coherent. (In fact, only the Emerald line and the K-Series are.)
- Drivers need to be modified so that they detect the existence of Coherent I/O on the system on which they are running in order to ensure optimal performance on both types of systems.

Coherent I/O features

- General:
 - Allows cache move-in rules to be relaxed without excessive software management
 - Removes necessity for drivers to perform cache flushes and purges
- The service `dma_sync` replaces all previous cache flushing, purging, and synchronization I/O services
- Existing DMA services now use coherent I/O, avoiding massive recoding of drivers

Notes

- Drivers for these platforms need to adopt the smart services, which determine the hardware type and act appropriately. Old drivers will suffer poor performance if you simply recompile the driver. If the driver uses the new services, it will be insulated from cache manipulation.
- The drivers can't tell right now if they are running on coherent I/O hardware; the driver developer must know the target environment and compile accordingly.

Summary and review

- **Modular architecture**
 - More flexible, open
- **Pre-10.0 platform differences persist**
 - Plan is to merge in future release
- **Significant consistency between S700 and S800**
 - Special file naming
 - Major and minor numbers converged
 - Administrative utilities
 - Disk layouts
- **Driver changes required**
 - All S800, most S700 drivers
 - Smarts for Coherent I/O support

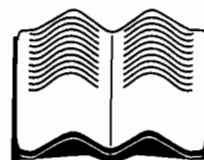
Notes

For more information



For more information

- *Configuring HP-UX for Peripherals*
- *Release Notes for 10.0*
- *HP-UX Driver Development Guide*



Notes

-
1. True or False: `/dev/null` is the only device that has the same file name as previous releases.
 2. Which changed at 10.0, major numbers or minor numbers?
 3. If you need to make a device special file, what's the preferred way for customers? (Hint: It's got a GUI.)
 4. Answer this customer call: "I need to create a special file from an installation script. What command should I use?"

Extra credit: Explain the advantage of that over the `mknod` command.

-
1. Explore the `/usr/conf/master.d/` directory. Do an `ls` on the directory and look at the files. What three-character sequence ends each driver table?
 2. Run `ioscan` without options to discover how many devices you have. For the S800, use the one from which you installed.

S700	S800
SCSI devices	
bus converters	
bus adapters	

3. What's the name of the Parallel printer driver on the S700? (Hint: Use `ioscan -f`.)
4. In which two directories are the device special files for tty devices? Use `ioscan -n`, although you will need another option besides `-n`. Try this exercise on the S800.
 - 1.
 - 2.
5. What's the major number of the driver for the tty devices? (The display is more interesting on the S800.) Also record the command that reveals the answer.

-
6. On the S800, look again at the output from the previous question. What's the difference in the three types of tty device files? Choose all that apply.

- ☐ Major numbers
- ☐ Minor numbers
- ☐ Other differences
- ☐ No differences

What do the differences in minor numbers mean for the tty devices? Run `lsdf` to find out.

7. Run `lsdf` on `/dev/sad`, `/dev/echo`, and any other interesting-looking files in `/dev/`.

`/dev/sad:`

`/dev/echo:`

8. The system initializes itself from the `/etc/inittab` file. Is the `iocinit` command mentioned there? (Hint: `grep iocinit /etc/inittab`)
9. Okay, so `iocinit` apparently isn't run directly from `/etc/inittab`. However, what command (from the `grep` output) probably does run it?
10. Look at that script (from #9). If `/etc/ioconfig` doesn't exist, in what two ways can it be created?
- 1.
 - 2.

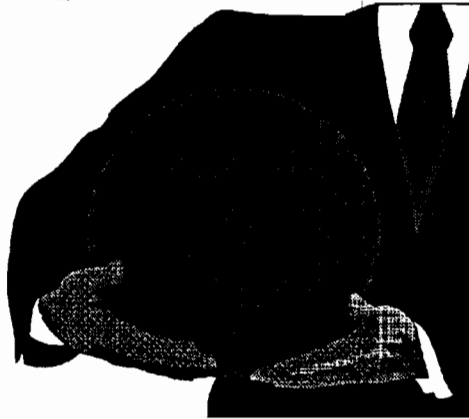
-
11. Based on the contents of the file from question #10, under what conditions might a customer see this shell prompt? _____

(in ioinitrc) # _

12. Put /opt/deltaclass/bin in your \$PATH if it isn't there already, then run iosscanner. This script uses the awk option of ioscan, piping the output to /opt/deltaclass/lib/iosscanner.awk, an awk script that you can read.

This page intentionally left blank.

Logical Volume Manager (LVM)



Module goals:

The purpose of this module is to explain how LVM changed, and show how to migrate customers of obsolete products to new LVM offerings. This module is *not* an LVM tutorial.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. List features of LVM that are new to this release.
2. Differentiate between the bundled and the optional products.
3. Recognize products that are obsoleted by new LVM features.
4. Migrate a non-LVM disk under 9.0 to LVM under 10.0.
5. Migrate a software disk striping (SDS) disk to LVM.
6. Migrate a DataPair disk to LVM.
7. Tell how (and why) to set up a read-only volume group.
8. Split a logical volume into multiple parts atomically.

Motivations for LVM changes

- **One HP-UX: Same on S700 as on S800**
 - Same commands and arguments
 - Absorb similar but proprietary products (like SDS on S700, DataPair on S800)
 - Provide mirroring on S700
- **Roll specials & patches into bundled product**
- **Make LVM commands smarter and easier to use**
 - Commands that change the configuration back up the change (vgcfgbackup, lvinboot)
- **Provide history log of what happened (syslog)**

Notes

- The auto-backup of LVM commands was in response to feedback by Response Center engineers.

Features and benefits

Features

- Support on S700
- V.4 file system layout for commands
- Automatic configuration backups
- Multi-computer access to the same disks

Benefits

- One HP-UX
- Follows modern standards
- Allows creation of more highly available systems

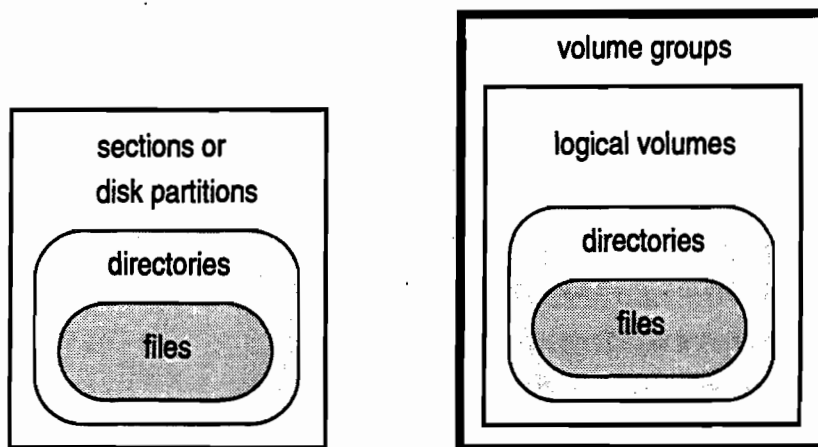


Notes

- High-availability features (explained in more detail later):
 - Physical volume links
 - Multiple computers can share disks
 - Read-only volume groups
 - Atomic multiple `lvsplit`

Partitioning vs. Logical Volumes

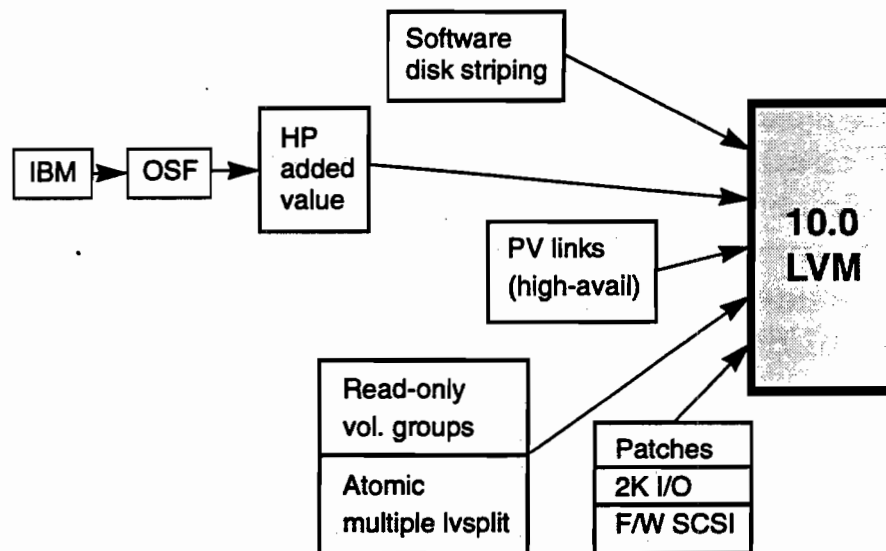
- Conventional partitioning vs. LVM



Notes

- With LVM, file systems and files can span disks. As before, a file system can be at most 4 Gb in size, and a file can be up to 2 Gb.
- With LVM, the system administrator controls the sizes of the logical volumes.

History of LVM-related features



Notes

- This LVM implementation has its roots at IBM. IBM actually got LVM from a third party. After giving it to OSF, IBM continued to develop it separately — so IBM's LVM is slightly different from everyone else's.
- Much of the functionality that is now part of 10.0 was at one time part of a special, or another implementation (i.e., SDS).
- Physical volume links (PV links) are sometimes called *alternate links*.

Overview of new features

- **Supported on S700**
- **Automatic config backups & boot info updates**
- **Physical volume links**
- **Files and commands follow V.4 convention**
- **Atomic multiple `lvsplit`**
- **Features for multiple computers sharing disks**
 - **Read-only volume groups**
 - **MC/ServiceGuard (optional S800-only product)**



Notes

- **Each of these topics is covered in more detail on the following slides.**

LVM now supported on S700



LVM now supported on S700

- All features same on S700 and S800
 - Includes root, swap, dumps
 - Includes mirroring
 - Includes bad block relocation
 - Commands and args are the same on both
- LVM on a particular architecture (S700/S800) will support all the disks of that architecture
 - Example: If a new disk appears only on S700, only the S700 version of LVM will support it
- Can `vgimport/vgexport` between S700 and S800, and between 9.0x and 10.0

Notes

- The LVM code contains only a couple of `#ifdefs` so the S700 and S800 should be feature-for-feature compatible. The one exception is differences motivated by different driver behavior.

Automated configuration backups

- Commands that change the configuration now perform the change and then run `vgcfgbackup` and/or `lvlnboot` if necessary
 - When the configuration is automatically backed up, it's logged to the `syslog` as well
 - These commands may still be run by hand, too
- As always, `vgcfgbackup` and `lvlnboot` back up LVM data structures, ***not user data!***

Notes

- Why might someone manually back up his configuration? One scenario:

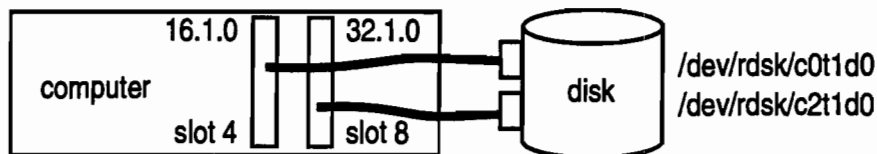
```
# vgcfgbackup -f /etc/lvmconf/vg06.save vg06
... lots of configuration-changing commands here
# vgdisplay -l vg06
... well, that didn't work; put it all back
# vgcfgrestore -f /etc/lvmconf/vg06.save vg06
```

LVM is not designed to work this way, but some will use it this way.

- Another scenario is a script that performs many `lvextend` commands. Each `lvextend` will auto-backup, slowing the script down. The `-A n` option turns off auto-backup, but be sure to do the manual backup at the end of the script.
- Of course, `syslogd` must be running for `syslog` to work (the system must be in a multi-user state).

Physical Volume (PV) links

- One disk, but dual links (hardware paths: bus, cards in the computer, cables) to it



- If one physical link fails, LVM re-routes the I/O to the surviving link
- HW limit: Max of 8 links to a single non-boot disk
- SW limit: Max of 4 links to *all* boot disks, meaning 1 boot disk can have 4 links, or else 2 boot disks in the root volume group can have 2 links each

Notes

- PV links guard against hardware problems like an I/O board going bad, or a SCSI cable getting pulled out, or dropping a floor tile on your FL cable. Because there is still a single point of failure (the heads, the platter, the controller in the disk drive), this is not a *fault-tolerant* solution. It is a *high-availability* solution.
- PV links work with SCSI, FL, and RAID disks. Most customers will probably use PV links with RAID disks.
 - RAID disk arrays handle features like striping and mirroring in the hardware. The array controller fools the rest of the system into thinking that its many disks are one large disk.
- No special commands or pvcreate options are necessary; LVM examines the PVID, which it assigns, and recognizes the PV link when it discovers a matching PVID.
- Be careful, the controllers must also have different SCSI targets, since they're on the same bus.

PV link commands

- **Create a new volume group**
pvcreate /dev/rdisk/c0t1d0
vgcreate /dev/vg01 /dev/rdisk/c0t1d0 /dev/rdisk/c2t1d0
- **Add an alternate link to an existing primary PV**
(vg01 has a PV with a link at /dev/dsk/c0t1d0 already)
vgextend /dev/vg02 /dev/dsk/c2t1d0
- **Removing an alternate link**
vgreduce /dev/vg01 /dev/dsk/c2t1d0
- **Removing a primary link**
vgreduce /dev/vg01 /dev/dsk/c0t1d0
(c2t1d0 is now the primary link to that disk)

Notes

- To create a new volume group, pvcreate only one of the links. Then vgcreate specifying all link device-names at once. In the example above, the disk at c0t1d0 is the primary link, and the disk at c2t1d0 is the alternate link.
- If the volume group is activated, use vgextend to add an alternate link. Do not pvcreate the new link.
- To exchange the primary and alternate links:
 1. Remove the primary link from the volume group, which makes the alternate link become the primary:
vgreduce /dev/vg01 /dev/dsk/c0t1d0
 2. Add the old link back in as an alternate link:
vgextend /dev/vg01 /dev/dsk/c0t1d0
- **Note:** As the system is rebooting, LVM chooses the first link in the /etc/lvmtab file as the primary link. Across reboots, it does not maintain information about which link was the latest primary.

Viewing PV links

- Use the -v option of vgdisplay:
vgdisplay -v /dev/vg04
- Physical volumes area will display “Alternate Link” for linked device

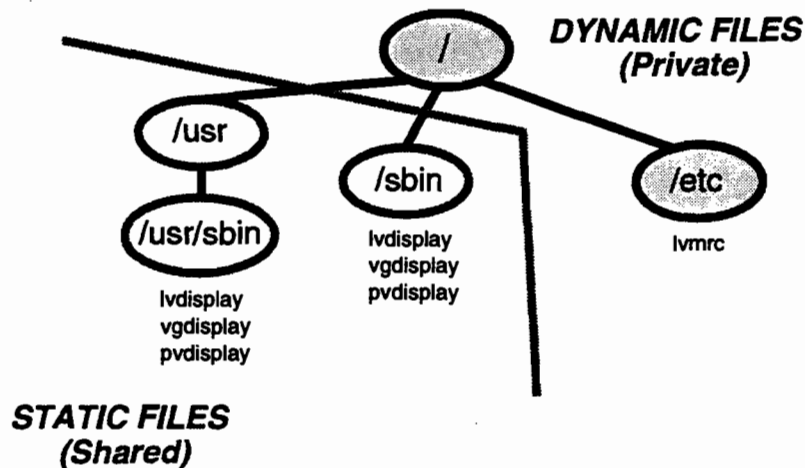
Notes

- Sample output from vgdisplay -v /dev/vg04:

```
--- Volume groups ---
VG Name                /dev/vg04
VG Status               available
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 16
Cur PV                2
Act PV                 2
...
--- Physical volumes ---
PV Name                /dev/dsk/c5t0d0
PV Name                /dev/dsk/c7t0d0 Alternate Link
PV Status               available
Total PE               544
Free PE                490

--- Physical volumes ---
PV Name                /dev/dsk/c6t0d0
PV Name                /dev/dsk/c8t0d0 Alternate Link
PV Status               available
Total PE               544
Free PE                544
```

V.4 layout for files and commands



Notes

- Commands all return a sensible status:
 - 0 for success
 - 1 for failure
- In each directory, there is one single LVM command binary.
 - Disk space usage went from 3 Mb (= 3000 Kb) to 300 Kb.
 - On a running 10.0 system, type:

```
$ ls -li /sbin/*display
```

```
$ ls -li /usr/sbin/*display
```

```
$ file /sbin/*display /usr/bin/*display
```

Then discuss in class:

 - How many LVM commands are there?
 - Why do the commands appear in both directories?

Commands for physical volumes

Physical volume	Host file system
<code>pvdisplay</code>	<code><i>newfs</i></code>
<code>pvcreate</code>	<code>extendfs</code>
<code>pvchange</code>	
<code>pvmove</code>	

Notes

- To initialize a disk or root disk, use the same steps as before.
- The `newfs` command changed at 10.0, but not specifically for LVM. This module will explain in more detail later.

Commands for VGs and LVs

Volume group	Logical volume
vgdisplay <i>(offline backup)</i>	lvdisplay <i>(disk striping)</i>
vgcreate	lvcreate <i>(disk striping)</i>
vgchange <i>(offline backup)</i>	lvchange
vgreduce	lvreduce
vgremove	lvremove
vgextend	lvextend
vgexport	lvmerge
vgimport	lvsplit <i>(atomic sync)</i>
vgsync	lvsync
vgscan	lvlnboot
vgcfgbackup	lvrmboot
vgcfgrestore	lvmmigrate

Notes

- The **vgdisplay** command also changed for PV links.

Atomic multiple `lvsplit`

- Allows synchronous split of multiple LVs
 - Useful for applications that split related data across volumes
- The `lvsplit` command takes names of up to 127 logical volumes on the command line
- Example

```
# lvsplit /dev/vgdb/data /dev/vgdb/logs
```

Notes

- After you type the `lvsplit` command, the LVM software:
 1. Freezes all transactions
 2. Syncs the LVs
 3. Splits the LVs
 4. Unfreezes the transactions
- For example, a database is using `lv2`, `lv3`, and `lv4`. You can split them at the same time (“atomically”) with:

```
# lvsplit /dev/vgdb/lv2 /dev/vgdb/lv3 /dev/vgdb/lv4
```

Multiple computers sharing disks

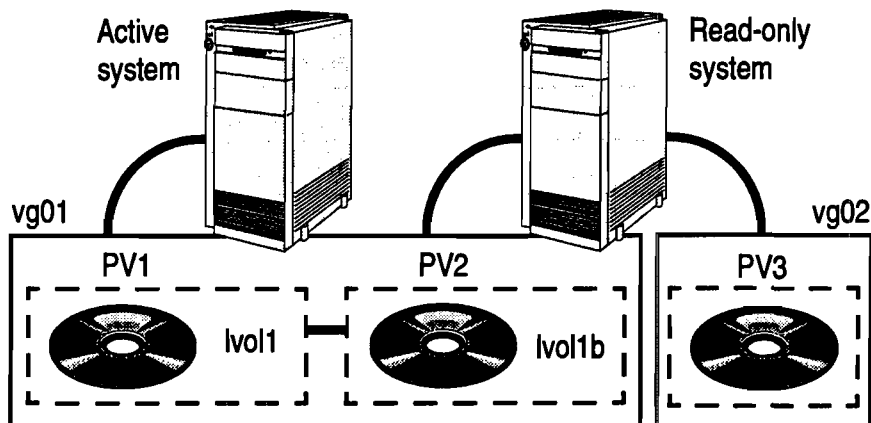
- Old 9.0x LVM did not do multi-computer access
 - Computers share disks this way through a direct connection, not the net — don't confuse this with NFS
- New capabilities to guarantee that boot and data disks are accessed by only one system at a time:
 - Read-only volume groups (for online backups)
 - Atomic multiple `lvsplit` (for online backups)
 - Disk locking through optional product MC/ServiceGuard

Notes

- The "MC" prefix identifies high-availability products.

Read-only volume groups

- Allows two systems to “safely” access same LV
- For online backups with no performance hit



Notes

- On the read-only system, LVM guarantees that the configuration and user data cannot be modified. To activate a read-only volume group:
`# vgchange -a r /dev/vg01` *# instead of: -a y*
- Restrictions:
 - Read-only volume groups are recommended only for online backups.
 - You cannot read and write the same logical volume from two systems at the same time. LVM doesn't support locking. (Optional product MC/ServiceGuard does lock, however.)
 - The logical volumes in the read-only group cannot be mirrored. The read-only system is unable to mirror the data.
- If the customer wants online backups, the optional product OnLine-JFS (available post-10.0) may be more appropriate. OnLine-JFS tools provide a better backup solution because they can be smarter than LVM. On the other hand, LVM is bundled with the system.

Example - Using lvsplit & "ro" VG

On the active system

```
lvsplit /dev/vg1/lv1
```

```
fsck /dev/vg1/rlv1
```

```
lvmerge
```

On the read-only system

```
vgchange -a r /dev/vg1
```

```
mount /dev/vg1/lv1 /bk
```

```
fbackup . . .
```

```
umount /bk
```

```
vgchange -a n /dev/vg1
```

Notes

- During this whole operation, the database application (or whatever happens to be using vg1 on the active system) continues uninterrupted.
- The fsck is required because of unnamed pipes and unlinked-but-open files on the active system.
 - For instance, one process is doing ping host >> /dir/ping.out. Then someone types rm /dir/ping.out. The ping process is still writing to the file even though it is removed; the result is a dangling inode. The fsck will clean up such problems.
- You can't back up memory-mapped files using this method. Memory-mapped files don't go through the buffer cache, so they are unaffected by these procedures.

Old features preserved

- S800 LVM has the same functionality as before
 - Scripts will still work unless they have hard-coded path names
- SAM understands LVM
- MirrorDisk/UX remains a separate optional product and is available on S700

Notes

- The LVM major number is still 64 at the 10.0 release.
- SAM continues to work as before:
 - It handles the easy stuff, so steer customers toward SAM. But don't mix SAM and shell commands for a single task.
 - SAM still lacks `lvreduce`; you must do that from shell commands. That command is considered too dangerous to automate.

Old features replaced

- **Software Disk Striping (SDS) from the S700 is obsolete, replaced with LVM disk striping**
 - **Some cases remain supported under the supervision of the I/O compatibility driver (cpd)**
- **DataPair is obsolete and unsupported at 10.0, replaced with MirrorDisk**

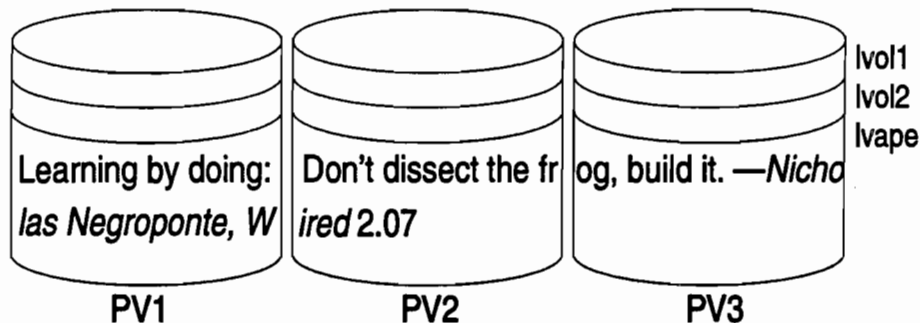
Notes

From SDS to LVM stripes

- Striping spreads data across multiple disks for high availability:

data:

Learning by doing: Don't dissect the frog, build it. —Nicholas Negroponte, *Wired* 2.07



Notes

- To create a logical volume striped across 3 disks, with a stripe size of 32 Kb, for a total LV size of 24 Mb:

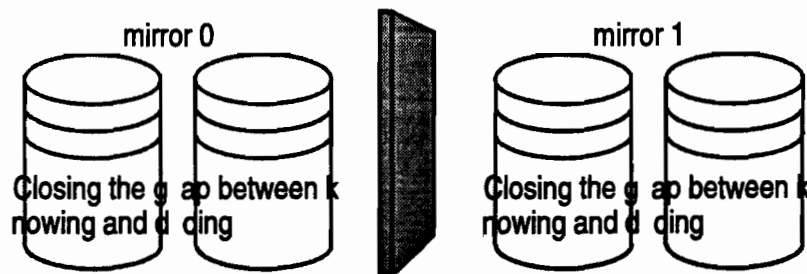
```
# lvcreate -i 3 -I 32 -L 24 -n lvape /dev/vg01
```

 - i Number of physical volumes
 - I Size of the stripe in KbThe other arguments are the same as in the non-striped case.
- Stripes go across physical volumes, from two up to the maximum number of physical volumes in the volume group. Valid stripe sizes are 4, 8, 16 (default), 32, and 64 Kb.
- It's OK to use RAID disks with striped LVs. LVM believes what it gets from the driver, so it doesn't know that the "disk" is an array of them.
- If PV2 above goes dead, your I/O can hang. Stripes are not fault-tolerant.

Quote used by permission of Nicholas Negroponte and *Wired* magazine.

Mirroring and striping (*post-10.0*)

- `lvcreate` will allow creation of striped/mirrored disk at a future release
- Parallel allocation policy only
- Same mirror used for all read requests
 - Allows for I/O chaining
- You can stripe something and mirror it later



Notes

- To create a logical volume striped across 3 disks, with a stripe size of 32 Kb, for a total LV size of 24 Mb, singly-mirrored:

```
# lvcreate -i 3 -I 32 -L 24 -m 1 -n lvglass /dev/vg01
```
- This is much simpler than under 9.0, where the `lvextend` was often scripted.
- The `lvsplit` command will split the mirrors in half.
- With mirroring you get roughly 2% improvement on read times.
 - LVM picks the shortest I/O queue.
 - Mirror Write Cache (MWC) can hurt write performance if writes are all non-sequential. This is easy to turn on and off, so take measurements and use the best setting in your circumstances.
- ***Mirrored striped volumes will be available at a future release.***

SDS to LVM migration tools

- *Single-partitioned* SDS disks (data striped on one disk) are supported on 10.0 via the cpd “hard partition compatibility” driver
 - Not a common case
- SDS *arrays* are automatically converted at upgrade time to LVM volume groups
 - ***Upgrade is not available at the 10.0 release***
- SDS on the root disk requires reinstalling
 - This configuration was never supported (we kept telling people not to do that)

Notes

- How arrays are converted during an upgrade (*not available at 10.0*): Data is not moved. The disks are simply converted over to LVM.
 - During the upgrade, after `/dev/` is converted, LVM scans all disks on the system and issues the `/sbin/sdstolvm` command to convert the SDS disk arrays to volume groups by rewriting the labels.

Migrating from DataPair to LVM

- Replacements:
 - Atomic multiple `lvsplit` and read-only volume groups (for online backups)
 - Optional product MirrorDisk/UX, which is LVM with mirrors
- You must unconfigure DataPair (or convert to MirrorDisk) before you upgrade to 10.0, perhaps the day before
 - The process is described in the 9.0 S800 *System Administration Tasks* manual, chapter 8, near the end of the chapter

Notes

- A logical volume can be accessed read-write from only one system at a time.
 - To do an online backup, use read-only volume groups and atomic multiple `lvsplit` as mentioned before
- Disk “locking” (really prevention of multiple simultaneous access) is handled at the granularity of a volume group through read-only volume groups.
- Sales of DataPair were not overwhelming, so migration should not be an overwhelming issue. Marketing has sent many letters to DataPair customers to alert them.

Migrating from non-LVM disks

- At 9.0, the S800 used fixed disk sections

OK at 10.0

You can mount file systems from existing sections

You can recover a fixed-section disk that crashed

You can replace a crashed disk with one of the same type

Not OK at 10.0

You cannot create fixed sections on a disk that lacks them now

You cannot repartition a disk

- Other changes: Section 2 is now 0; newfs arg; names in /dev/; major & minor numbers

Notes

- From a 9.0x S800 root disk to LVM:
 - **Recommended:** Install 10.0, converting the root disk to LVM in the process, and then restore your 9.0 user files. The install tools are smarter than `lvmigrate` in the matter of V.4 file locations.
 - Another install option is to avoid LVM to go with whole-disk access through section 0, but this is not recommended because you'll lack LVM.
 - **Upgrade**, converting either before or after with `lvmigrate`. You cannot convert during an upgrade. *Upgrading is not available at the 10.0 release.*
- From a 9.0x S700 disk to LVM:
 - **Recommended:** Install 10.0, converting the root disk to LVM in the process, then restore your 9.0 user files.
 - **Upgrade**, then convert with `lvmigrate`. *Upgrading is not available at the 10.0 release.*

Tips and tricks

Customer call

Your reply involves

I just installed 10.0 and my DataPair mirrored disks are not recognized any more! How can I convert them?

MirrorDisk/UX optional product

Can I restore my 9.0 LVM configurations onto my 10.0 system and use those?

Yes, use vgexport/vgimport

I want to back up the files of my users. How can I lock the disk for the /home/ directory while I run fbackup?

Disk "locking" at the VG level, read-only VGs, lvsplit, another system required

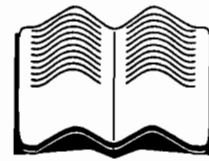
Notes

For more information



For more information

- *How HP-UX Works: Concepts for the System Administrator*
- *System Administration Tasks*
- *Release Notes for 10.0, HP 9000 Series 700/800 Computers*



Notes

- *How HP-UX Works: Concepts for the System Administrator*
 - Chapter 9, "Logical Volume Manager," is a tutorial to LVM.
- *System Administration Tasks*
 - Chapter 3 discusses LVM; chapter 4 examines file system issues; chapter 6 explores swap space; chapter 7 looks at mirrors; chapter 8 details striping.
- *Release Notes for 10.0*
 - In chapter 3, "Major Changes for HP-UX 10.0," the section entitled "Disk Management" talks about changes to LVM and gives suggestions for converting SDS and fixed-section disks to LVM.
 - Chapter 6, "Commands and Libraries," contains an alphabetical list of changes that includes newfs (1M).

-
1. List features of LVM that are new to this release.
 2. Since a version of LVM is bundled with HP-UX, why would someone buy the extra product?
 3. What products are obsolete because of new LVM features?
 4. Why would someone set up a read-only volume group?

For each activity, record the commands you used in the space beneath the question or in the margins. If necessary, your instructor will give you any special instructions needed to perform this lab.

1. Create a "control" logical volume of 72 Mb. Use the defaults for parameters whenever possible, except give it a name of `lvcontrol`. Run `newfs` and mount the file system on a directory called `/control`.

2. Create striped logical volumes. Create a new 72-Mb logical volume named `lvstripe` that is striped across two physical volumes.

If you have the hardware, create another 72-Mb logical volume named `lvplaid` striped across three physical volumes.

3. Verify that the data will really be striped by asking LVM for information about the logical volume(s).
4. Make a new file system on the volume(s) and mount them. Mount `lvstripe` on a directory named `/stripe`, and (if you can) `lvplaid` on a directory named `/plaid`.
5. Write some large files to the control logical volume. How long does it take to write to it?

How long does it take to write to the striped logical volume(s)?

6. How long does it take to read the large file from the control volume? From the striped volumes?
7. Remove the striped logical volumes, but keep the control volume.
8. Now reuse the same disk space as before to create 72-Mb volumes that are mirrored. Create a new 72-Mb mirrored logical volume (if you can) named `lvmirror` that is mirrored once (one mirror plus one original).

If you have the hardware, create another 72-Mb logical volume named `lvfunhouse` that is mirrored twice.
9. Verify that the data will really be mirrored by asking LVM for information about the logical volume(s).
10. Make a new file system on the volume(s) and mount them. Mount `lvmirror` on a directory named `/mirror`, and `lvfunhouse` on a directory named `/funhouse`.

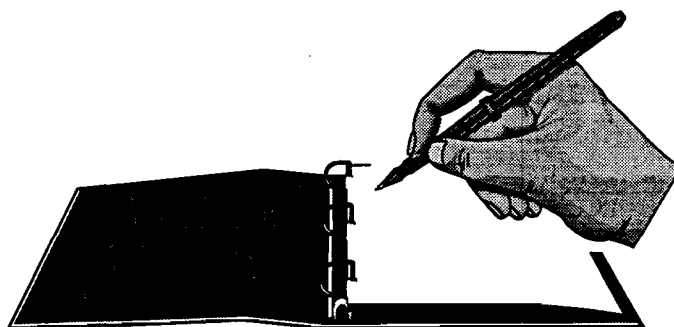
-
11. Write some large files to the control logical volume.
How long does it take to write to it?
-

How long does it take to write to the striped logical
volume(s)?

12. How long does it take to read the large file from the
control volume? From the mirrored volumes?
13. Remove the mirrored logical volumes. Remove the
control volume, too. Don't forget to remove the mount-
point directories as a final step.
14. Look at the syslog system log file. Which messages
are from LVM? What kinds of events trigger them?

This page intentionally left blank.

Journal File System (JFS)



Module goals:

The purposes of this module are to discuss the administration of a journaled file system (JFS), to describe the JFS layout, and to discuss the tuning of JFS.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the features and benefits of JFS.
2. Indicate the differences between OnLine JFS and Base JFS.
3. List and describe the components of JFS disk layout.
4. Use the JFS administrative commands.
5. Tune a JFS for optimum performance and reliability.

Motivations for providing JFS

- Premier high-end file system from Veritas
- Proven file system
 - It already runs on other vendors' systems
 - IBM, USL
- Enables customers to have more uptime and better reliability and performance
 - Fast file system recovery
 - Purchasable product has on-line administration features and more performance improvement configurability

Notes

- History
 - Tolerant Systems (company) became VERITAS
 - AT&T requested a file system that could be turned on and off like DOS
 - Standard in SVR4.2
- NOTE: This is not available with HP-UX 10.0, but will be available at a future release.

Base JFS vs. OnLine JFS features

- Base JFS comes with HP-UX
- OnLine JFS is purchasable
- Both have:
 - Fast file system recovery
 - Extent-based allocation
 - Enhanced mount options
- OnLine JFS has all Base JFS features and more:
 - On-line administration and backup
 - Caching advisories
 - Extent attribute modification
 - File access advisories
 - Intent log disabling



Notes

- In this module, we will cover all the features contained in both first, then discuss those features supported only in On-Line JFS
- Note on product names:
 - Marketing calls these products (the official names) as above.
 - Engineering, and therefore all the software commands, refer to these products as:

The Base VxFS File System vs. The Advanced VxFS File System
Both are known as “vxfs” or “VxFS” for short, by the lab
Commands and ioctl calls specific to JFS/VxFS are prefixed by vx.

Fast file system recovery

- JFS logs pending changes to an *intent log*
- JFS `fsck` utility reads intent log and repairs file system based on it
 - Nullifies or completes file system operations that were active when system failed
- Completes recovery in a few seconds
 - Scans intent log, not entire file system
- Intent log recovery feature is transparent to system administrator and users
- Old style full checking still available
 - Must run to recover from disk hardware failure



Notes

- If there is a write-on-disk error or an I/O error, a bit gets set to run full `fsck`.
- Intent log is described in more detail later in this module.

The intent log

- Circular activity log
- Transactions are marked as they are completed, and eventually get overwritten
- A transaction is an update to the file system
- For example, a transaction that creates a file and expands the directory requires these 5 steps:
 1. Update the free extent map for directory to increase directory size.
 2. Change the allocated block.
 3. Modify the directory inode to reflect new size.
 4. Modify the inode for new file.
 5. Update the free inode map.
- What if it crashes before those steps complete?

Notes

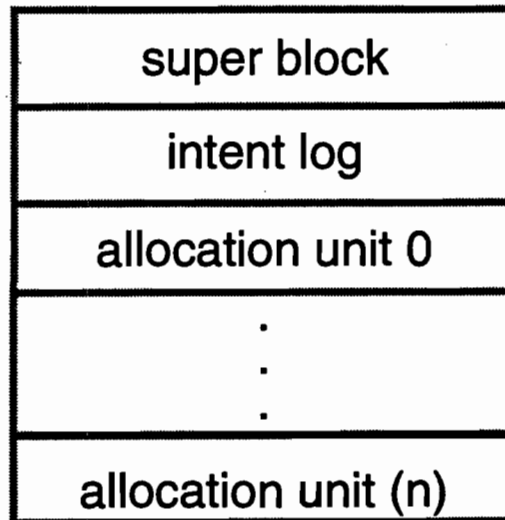
- The system could crash before all five steps listed above have completed, leaving the file system in an inconsistent state.
- The intention to perform these steps is logged before the file system is changed.
- If the system crashes before all steps are complete, `fsck` can read the intent log and reconstruct the file system without performing an exhaustive analysis of the file systems' contents. Thus, the amount of time that it takes to `fsck` a JFS is dependent on the size of the intent log, not the size of the file system. Since the intent log is small, `fsck`'s complete very quickly.
- The intent log size is configurable.
 - Use the `mkfs "logsize"` option.

Extents & extent-based allocation

- When storage added to a file on JFS, it is grouped in extents, not blocks
- Extents are:
 - Contiguous file system blocks treated as a unit
 - Incremental unit of allocation used when file is to be extended
- Allows for I/O in multiple blocks which is much faster than block-at-a-time operations

Notes

JFS layout



Notes

- The layout of a JFS is similar in concept to that of an HFS. Allocation units replace cylinder groups in this model, though.
- At a high level, the JFS superblock contains the same kind of information as an HFS superblock. The exact structure is different.
 - Programs that read the superblock of a file system directly will not work on both the HFS and JFS without modification, since HFS and JFS superblocks are not identical.
- Copies of the superblock are kept in allocation unit headers so the system can be recovered if the primary one is destroyed.
- The rest of these elements are described in more detail on the following pages.

The allocation unit (AU)

allocation unit header
free resource summaries
free inode map
extended inode operations map
free extent map
inode list
padding
data blocks

Notes

- Similar to an HFS "cylinder group."
- The number and size can be specified when the file system is made.
- All allocation units, except possibly last one, will be of equal size.
 - Last allocation unit can have a partial set of data blocks to allow use of all available blocks on a partition
- To align data blocks to a physical boundary, can specify gap to be left between the end of inode list and the first data block
 - This is padding

Components of an AU

- *Allocation unit header* contains copy of superblock
- *Free resource summaries* contain raw counts of inodes with extended operations pending, free inodes, free extents in allocation unit
- *Free inode map* is bitmap indicating state *and* location
 - So is *extended inode operations map*
- *Free extent map* holds location and size of extents that can be allocated to files
- *Inode list* is the list of inodes (one per file) for each file in file system
 - Size of list is dynamically determined

Notes

- Copy of superblock can be used if primary superblock is destroyed.
- Free inode is indicated by bit in map being “on”.
- Extended inode operations map stores location and state of inodes with pending changes that have indefinite completion time (e.g., last unlink of open file).
- The inode list is used for dynamic inode allocation.

Old commands that know JFS

Command	Purpose
<code>ff</code>	make pathname list
<code>fsck</code>	check and fix file system
<code>fsdb</code>	debug and patch file system
<code>labelit</code>	label file system
<code>mkfs</code>	make file system
<code>mount</code>	mount file system
<code>ncheck</code>	pathnames and inode list
<code>volcopy</code>	copy file system

Notes

- To access JFS version of a command, use:
`command -F vxfs`

Commands new with JFS

Command	Purpose
getext	display extent attributes
vxdump	back up JFS
vxrestore	restore backup

Notes

- The vxdump utility is very similar to dump.
- The vxrestore utility is very similar to restore.
 - Additionally, it provides for preservation of extent attributes of files in its source archive, provided the restore is done onto a JFS.
- Basic extent attributes associated with a file:
 - Reservation (reserved space)
 - Fixed extent size
- Cannot modify extent attributes with Base JFS.

Performance and tuning

- **Modify block size for the file system**
 - Default is 1K
 - No fragments
- **Modify size of the intent log**
 - Default is usually 512 blocks
 - If file system is < 4MB, it is made smaller
 - Larger size may enhance performance with NFS
- **Modify file system mount options**
 - Use new mount options for JFS
- **Tune kernel variables**
 - As for other file systems

Notes

- Use `mkfs bsize` option to modify block size.
- Use `mkfs logsize` option to modify intent log size.
- Regular HFS files use larger block sizes for better data transfer rates.
- On HFS, block is fragmented for use by multiple files, so as to not waste space.
- JFS keeps files contiguous allowing large transfers of data.

New mount options

Option	Purpose
log	default-- logs activity (intent log)
delaylog	some delayed write of intent log
tmplog	delayed write of intent log
datainlog	default--logs synchronous writes
nodatainlog	for disks without bad block revectoring
blkclear	clear extents when unused
mincache=closesync	flush when file is closed

Notes

- Options to improve data integrity:
 - blkclear causes 0's to be written to file data blocks before use (slow, but more secure)
 - log causes intent log to be used
 - mincache=closesync reduces the system's exposure to data loss in the event of a crash, but will slow I/O to the file system down by 10-15%.
- Options to enhance performance:
 - delaylog increases amount of kernel buffering before intent log is written
 - tmplog further increases amount of kernel buffering and is only appropriate for file systems with "don't care" data

On-line system administration

- Works on mounted file system.
- Done with fsadm utility
- Defragmentation (reorganization)
 - Removes unused spaces from directories
 - Makes small files contiguous
- Consolidates free blocks
- Resizing
 - Allows resizing of file system while it is being actively used (mounted)

Notes

- In other file systems, an unmounted file system could be reorganized and defragmented using the dcopy utility, for example.

On-line backup and snapshots

- A *snapshot* of mounted file system is used for making backup
- Snapshot is constant view of file system “snapped” at point in time when snapshot is made
- *Snapshot file system* provides consistent image of snapped file system, with all activity completed or blocked at system call level
- Will slow down performance, as initial writes to a block on snapped file system now must:
 1. Read old data
 2. Write data to snapshot file system
 3. Write data to snapped file system



Notes

- Snapshot file system is an instantaneous read of entire file system.
- It exists only as long as it is mounted.
- *Snapped file system* cannot be unmounted until corresponding snapshots are unmounted.
- Multiple snapshots of file systems can be made.
- It is not possible to make a snapshot of a snapshot.
- To backup selected files, use standard utilities: `cp`, `cpio`
- To backup up entire file system image, use: `volcopy`, `vxdump` or `fscat`
- The `fscat` utility's output is useful for an application requiring random access to snapshot image.
- If `vxdump` is not used to perform complete backup, then run `fsck` on restored file system, as snapshot file system is consistent, not clean
- Snapshot file system is read-only, so `fsck` cannot be run on it.
- Use `snapof= suboption` to mount to create snapshot file system.

OnLine JFS: New commands

Command	Purpose
<code>fsadm</code>	administer JFS
<code>fscat</code>	concatenate raw file system
<code>setext</code>	set extent attributes

Notes

- The `fsadm` utility provides on-line administration facilities.
 - Resize file system
 - Report on extent or directory fragmentation
 - Reorganize extents or directories.
- The `fscat` utility outputs byte stream representing raw file system.
- Use `setext` to manipulate basic extent attribute of reservation.
 - Manipulate to preallocate space to file
 - Make a persistent or temporary attribute of file
 - Adjust size of file to include reservation (superuser only)
- Use `setext` to manipulate basic extent attribute of fixed extent size.
 - Can set to override default allocation
- Other controls on extent attributes determine:
 - Alignment of allocations
 - Contiguity of allocations

On-line JFS: new mount options

Option	Purpose
<code>nolog</code>	disable logging activity (intent logging)
<code>snapof=special</code>	mount fs as snapshot of <i>special</i> , which is block special file of a mounted JFS
<code>snapsize=size</code>	<i>size</i> in sectors of snapshot fs being mounted
<code>mincache=direct</code>	converts reads and writes to direct reads and writes
<code>mincache=dsync</code>	converts writes to data synchronous writes
<code>convosync=closesync</code>	converts synchronous writes to asynchronous
<code>convosync=direct</code>	converts synchronous I/O to direct I/O
<code>convosync=dsync</code>	converts synchronous writes to data synchronous writes

Notes

- Disabling intent logging altogether with `nolog` makes all I/O requests asynchronous. This means that the file system state after a crash is likely to be extremely corrupt. It is only appropriate for file systems (like `/tmp`) that have data which is not guaranteed to be preserved across reboots.
- The `snapof` and `snapsize` options are used to mount a snapshot file system.
- The `min*` directives modify the system's caching of file system data. Without these, the JFS file system follows "normal" buffer caching rules; write structural data synchronously and contents asynchronously. With `dsync`, file contents are written synchronously. With `direct`, the file system buffer cache is bypassed altogether, and it's up to the application to manage buffered data.
- The `convosync*` options change the behavior of writes to files that have been opened with the `O_SYNCIO` flag. The `closesync`, `dsync`, and `direct` options have the same effect as previously described, but only on files that have been opened for synchronous I/O.
- Be careful in deciding when to use these options!

Things to know about JFS

- It cannot be used as rootable or bootable file system in first release.
 - This may be supported in the future.
- It may not be usable on systems with very limited memory as it is larger than HFS
 - 350K for Base JFS
 - + 40K for OnLine JFS (390K total for JFS)
 - 90K for HFS
 - Not including dynamically allocated memory
- It does not support ACLs.
- A trusted system will not include JFS.

Notes

- OnLine JFS does not replace Base JFS. It is an incremental product, in addition to Base JFS.

Tips and tricks

- How to determine if system has OnLine JFS?
`swlist | grep -i vxfs`
 - Reports both Advanced VxFS and Base VxFS on system if OnLine JFS is installed
- How to go from HFS to JFS?
 1. Backup HFS using supported backup utility, such as fbackup
 2. Use SAM to create JFS and mount it
 3. Restore files saved in step 1 (e.g., frecover)
- What applications may not work with JFS?
 - Those that are aware of HFS-specific information, such as HFS disk format

Notes

- In 10.01, SAM will support creating JFS and will allow use of file system administration utilities that support JFS.

Summary and review

- JFS improves uptime, performance, reliability
- JFS tracks intended changes to the file system by logging them in an intent log
- During system recovery, `fsck` scans intent log and makes any necessary changes--fast
- Extents, the file storage unit with JFS, are one or more contiguous blocks
- If OnLine JFS is used, some additional features:
 - Resize and defragment an active file system
 - Backup an active file system using snapshots
 - New `ioctl` calls and options to mount allow control of file system logging and caching

Notes

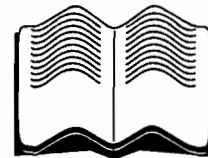
- Because the file storage unit can be multiple blocks, I/O with JFS can be considerably faster than with traditional file systems.

For more information



For more information

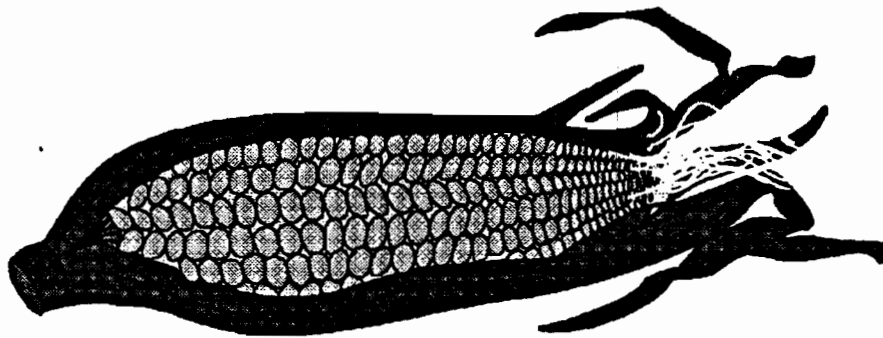
- The man pages for each command
 - Not available until JFS released
- *System Administration Tasks*



Notes

-
1. Why does JFS's fsck go so quickly?
 2. What is in an intent log?
 3. What is an extent? How does it help performance?
 4. What are the components of the JFS disk layout?
Briefly describe each one.
 5. What are three major differences between OnLine
JFS and Base JFS?
 6. How could you improve reliability on JFS?
 7. How could you improve performance on JFS?
 8. What is a snapshot file system?

Kernel Generation



Module goals:

The purpose of this module is to learn why and how to generate a new kernel. It also gives a glimpse of how the kernel has already been tuned by high-level prepackaged configurations.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Choose kernel generation options, generate a new kernel, install it, and boot it up.
2. List the reasons for and describe the mechanics of reclaiming disk space taken by kernel files. Describe how to reinstall the kernel files if these files are removed.
3. Configure some tunable kernel parameters.

Motivations for changing

- One HP-UX, one kernel generation process
- Make it easier to develop and add drivers to S800
 - Hard to port drivers to S800
 - S800 has a non-standard driver environment compared with S700, Sun, others
 - Not as open an architecture
- Support the J and K Series when they are released

Notes

- The factory also wanted to support some previously-undocumented procedures for recovering the kernel build file (A.K.A. the S800 file, the dfile, and now the system file).
- The K Series (Kittyhawk) is a S800 with a S700 CORE I/O card and a graphics top. To support it, we had to change the way we build kernels, among other things.

Features and benefits

Features

- Same system file and command to build a kernel
- Driver writing and installation is more like other Unix versions
- System file can be pulled out of kernel

Benefits

- One syntax is easier to learn and remember
- Open architecture — easier to get third parties to port their software to HP-UX
- No more rummaging around in file system or backup tapes for the current system file

Notes

- You could pull the `dfile` or `genfile` from the kernel under HP-UX 9.03, too, so this isn't strictly new.

Why generate a new kernel?

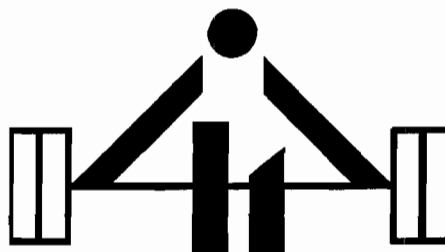
- Add device drivers or subsystems
 - Remove unnecessary drivers, too
- Tune the kernel
 - For performance
 - To support requirements for large applications
 - To support requirements for small systems
- Change location of devices used by kernel
 - Dump and swap
 - Console and root are no longer changeable in the system file like they were on S800

Notes

- More information on device drivers is in the manual *Configuring HP-UX for Peripherals*, new for 10.0.
- You cannot change all four sacred devices (root, console, dump, swap) as you used to be able to do under 9.0 on the S800. More on this later.
- For an example of a large application, Oracle 7 needs more shared memory. For an example of a small system configuration, HP-UX Lite scales down several tunable parameters to save memory.

Ways to rebuild the kernel

- SD-UX regenerates it on your behalf
 - If needed, when adding a new package
- Let SAM do it
 - *The preferred method for customers*
- Handle it manually with HP-UX commands
 - `/usr/sbin/sysadm/system_prep`
 - `mk_kernel`



Notes

- SD-UX can rebuild the kernel and reboot to increase shared memory segments, for instance. This class module doesn't look at the SD-UX case because it works without your intervention: Effectively, `swinstall` extracts the system file, modifies it, builds the kernel, and reboots.
 - At new major releases, the `swinstall` process removes all non-core drivers before rebuilding the kernel. The tunables, though, remain untouched.
 - Driver developers use functions in the customize scripts to add back their drivers. See the *Driver Developer's Guide* for more details.
- This module doesn't need to spend a lot of time on SAM because SAM takes you by the hand and leads you through the steps. *Encourage customers to reconfigure their systems with SAM.*

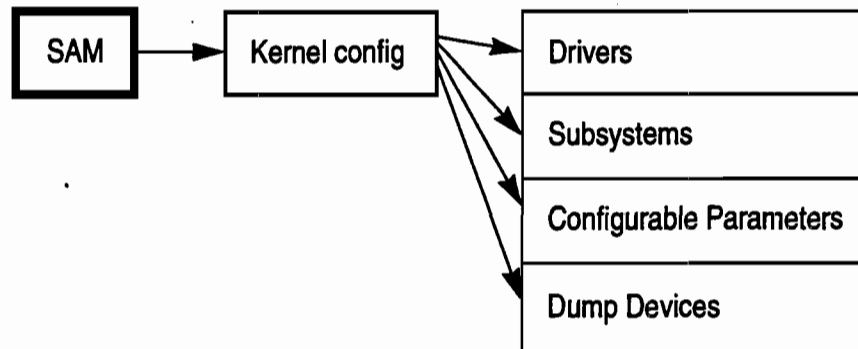
SAM's capabilities

- Set up primary swap device and dump device
 - On logical volumes
 - On whole disks
 - On hard partitions
- Modify primary swap device and dump device
 - When to enable (now, at bootup)
 - Swap priority (0 = use first)
- Change from a standard configuration to HP-UX Lite (and back again)

Notes

- The lab calls it the *dump* device, but the field call it the *dumps* device. SAM calls it the dump device, so let's be singular.
- SAM's *Kernel Configuration* area opens up to a *Configurable Parameters* icon. This has a menu item *Actions: Apply Tuned Parameter Set*, which brings up a dialog box. From this dialog box, you can select the settings for *Single-user commercial desktop workstations*, which is HP-UX Lite.
- HP-UX Lite trims down the kernel parameters and enables a minimal (core) set of critical subsystems.
 - Scales down several tunable parameters from their default values
 - Includes critical drivers (like SCSI disk, graphics for X11, NFS)
 - Omits non-critical drivers (like CD, SCSI tape, floppy, EISA, diagnostics, SLIP, printers)
 - Does not start non-critical subsystems or daemons (e.g., online diagnostics, network tracing and logging, the `ptydaemon` used by `vt` and `uucp`) in `/etc/rc.config.d`

SAM's kernel configuration area



Notes

- Also in SAM: Hard Disks, Arrays, CD-ROMs, and Floppies
 - Add a disk...
 - Using LVM ...
 - Define Swap LVs
 - Define File System LVs
 - Create or Extend a VG
 - Add New File System LVs
- Behind the curtains, SAM calls:
 - /sbin/ioscan To identify hardware and device files
 - /sbin/mksf To create device files
 - /sbin/insf To create device files

Manually generating a kernel

1. `cd /stand/build`
2. `/usr/sbin/sysadm/system_prep -s system`
3. `vi system`
4. `mk_kernel -s system`
5. `mv ../system ../system.prev &&`
`mv system ..`
6. `mv ../vmunix ../vmunix.prev &&`
`mv vmunix_test ../vmunix`
7. `cd /; shutdown -h 0`
8. Connect new devices, if any
9. Boot up

Notes

- The *Configuring HP-UX for Peripherals* manual contains a cookbook for rebuilding the kernel in nearly every chapter. Each recipe differs from the others only in details for setting up individual devices. For example, terminals require `getty` entries in `/etc/inittab`, but new LVM root disks require `lvlnboot` to set up root, swap, and dump.
- The full syntax for `mk_kernel` is:
`mk_kernel [-v] [-o outputKernel] [-s systemFile]`
- As the system reboots, it runs `ioscan` and `insf` to bind the driver chains in the kernel and make the `/dev/` file entries for the new device. To verify this, login and run `ioscan -fun [-C deviceClass]`.
- If the device is non-auto-configurable, you can associate a particular driver with the device by first creating an appropriate system file (in `/usr/conf/master.d`), creating and booting off of the new kernel), and running two commands:
`# ioscan -M deviceDriver -H hardwarePath`
`# insf -H hardwarePath`

Main ingredients in kernel gen

Directory	File	Contents
/stand/	system	List of ingredients in the running kernel
	vmunix	Running kernel
	ioconfig	Kernel data structure io_tree on disk
	bootconf	Location of boot LIF
/stand/build/	system	List of ingredients in the new kernel
	vmunix_test	Freshly-baked new kernel
	space.h	Pointers to sizes of kernel data structures
/usr/conf/- master.d/	*.h	Driver tables

Notes

- Hidden ingredients:

File	Contents
/usr/conf/*	Kernel headers and libraries
/usr/conf/space.h.d/*	Sizes of kernel data structures (referenced by /usr/conf/h/space.h)
/usr/conf/gen/config.mk	Default makefile template
/var/adm/sw/swagent.log	Errors and warnings from generation processes
/var/adm/sw/products/- prodName/filesetName/	Customize scripts

- See master(4), bootconf(4), ioconfig(4), and config(1M) for more information about these individual files.
- The mk_kernel command calls config, which calls make, but you are not supposed to know that.

Driver changes

Systems	Old name	New name	What
700 800	autoch	ssrfc	MO surface driver
700	autox	schgr	MO autochanger driver (still called autox on S800)
700	parallel	CentIf	Centronics/parallel port driver
700	scsi	sdisk	SCSI disk driver
700 800	—	cpd	Compatibility pseudo driver for SDS and hard disk partitions
700 800	—	kepd	Kernel entry point driver

Notes

- Obsolete drivers:

Systems	Obsolete	Replaced by	Notes
700	s2tape	scsi_tape	SCSI 2 tape driver functions
700	sds	LVM drivers	Software Disk Striping is obsolete
800	mirror	LVM drivers	DataPair is obsolete
800	pdisc	cpd	Superceded
800	gpio0		CIO AF/GPIO driver
800	gpio1		NIO GPIO LDM
800	gpio11		NIO GPIO DM

- Non-auto-configurable drivers are no longer configured in the kernel. Configure them without rebooting. Deconfiguration requires rebooting.

LAN driver changes

System	Obsolete	Replaced by	Where
	lan, lan01		
800		lan0	CIO (835, 870)
800		lan1	NIO Type A (815, 8x2, ...)
700 800		lan2	LASI, EISA (K Series, Wright Bros.)
800		lan3	NIO Type B (Nova 8x7, 890, ...)

- OK to include them all, but they will take up 120 Kb together

Notes

- LAN products now require STREAMS keywords and DLPI modules.
 - Keywords for STREAMS: hpstreams, timod, tirdwr, clone, strlog, sad, echo
 - SD-UX will insert the keywords into your system file at install/upgrade time, so they should be there already.
 - The Data Link Provider Interface (DLPI) is for low-level STREAMS-based access to network interfaces. It supports 802.3, 802.5, FDDI, and Fibre Channel.
- Driver developers:
 - See chapter 3 of the *Release Notes*, in the "Drivers" section, for a summary of how to convert your drivers.
 - See chapter 3 of the *HP-UX Pre-Upgrade Tasks Document* for warnings and procedures on coping with non-HP drivers.

Format of the system file

- Like S700 dfile
 - No root statement like on S800
 - No console statement like on S800
 - Only one swap statement allowed (sets primary)
 - * This is a comment line
- Sections, without separators:
 1. Drivers (names listed inside files contained in the /usr/conf/master.d/ directory)
 2. Optional section for dump and swap
 3. Optional section for tunable parameters

Notes

- Each file in the /usr/conf/master.d/ directory is like the S700 /etc/master file at 9.0x, except:
 - Tables begin with `$tableName`
 - Tables end with `$$$`
 - Tables may appear in any order, or may be omitted
 - New tables were added for 10.0
 - See master(4) for the details
- Some major numbers are now dynamic, but persisting across reboots.
 - In the future, new drivers will use dynamic major numbers, but some are still statically assigned in the files in /usr/conf/master.d
 - The `lsdev` command reports the major number for a driver — users of `mknod`, take note

New tunable kernel parameters

Parameter	Function	Meaning
create_fastlinks	Fast symlinks	Default = 0 (classic); non-zero = faster, avoiding one disk block access
dbc_min_pct, dbc_max_pct	Dynamic buffer cache	Amount of memory reserved for kernel buffers used in file I/O
streampipes	Pipes use STREAMS	Default is to use kernel buffers for pipes (classic; faster)

- Buffer Cache size is now variable on S700 & S800
 - Turn off variability by setting
dbc_min=dbc_max and specify desired %

Notes

- Defaults versus HP-UX Lite parameters:

Defs	Lite	Parameter name	Meaning
5	2	dbc_min_pct	Dynamic buffer cache minimum percentage
50	25	dbc_max_pct	Dynamic buffer cache maximum percentage
276	120	nproc	Maximum number of processes
476	200	ninode	Maximum number of open inodes in memory
671	350	nfile	Maximum number of open files
200	20	nflocks	Maximum number of file/record locks
60	30	npty	Maximum number of pseudo-terminals
50	20	msgmni	Maximum number of message queues
64	20	semmni	Maximum number of semaphore identifiers
200	50	shmmni	Maximum number of shared memory segments
30	1	ndilbuffers	Maximum number of DIL I/O buffers
100	70	scroll_lines	Maximum number of scroll lines on the graphics ITE
32	2	maxusers	Sets kernel data structure sizes (<i>not</i> number of logins)

Changing the console device

- Kept in Stable Storage (processor dependent hardware); set in ISL now, not in the system file
- 1. Shut down the system with SAM or shutdown
- 2. Boot to ISL, interacting with it
S700: Attended boot, then
Select from menu: **b P0 isl**
S800: Press any key within 10 seconds.
Then Interact with IPL (Y or N)?> **y**
- 3. From ISL, change the console path:
ISL> **conspath hardwarePath**
- 4. Turn off the machine, connect the new console
- 5. Boot up

Notes

- You can't do this from SAM, obviously.
- The checklist is in *Configuring HP-UX for Peripherals* in chapter 3, in the section on "Moving the System Console."
- Non-HP terminals can be used as the console, but you'll need a valid TERM setting before you can edit on it. As documented in chapter 3 of *Configuring HP-UX for Peripherals* in the section entitled "Configuring a Non-HP Terminal," this may involve:
 - tic and a valid terminfo file
 - Setting the environment variable TERMINFO
 - Setting the environment variable TERM

Changing the root device

- Issues: Exported NFS? LVM mirrors? In different volume group? AUTO file contains hardware path? /stand/system file specifies swap and dump? Whole disk, hard partitions? Boot LIF? ...
- First handle the issues, then change primary path in the boot ROM (and build a boot LIF)

Changing the path in Stable Storage

1. Reboot to ISL
2. Use `primpath hardwarePath`, as in:
ISL> **primpath 52.3.0**
3. Reboot:
ISL> **hpux /stand/vmunix**

Notes

- Moving the LVM root disk is documented in *Configuring HP-UX for Peripherals* in chapter 4 in the section on “Moving a Disk Drive to a Different Address.” This is a checklist containing twenty steps and even more sub-steps.

Setting the dump device

- Syntax in system file:
dump lvol
dump default
dump *hardwarePath*
dump *hardwarePath* s#
- Multiple dump devices OK (system uses last ones first; previous entries are used only if needed)
- Restrictions:
 - If the system file has no swap statement, you cannot use dump default
 - If dump device is defaulted to primary swap, SwitchOver must *not* be configured

Notes

- Note that the restrictions on the use of dump statements are the same as in previous releases.

Setting swap

Defining primary swap

- Device swap in system file:

swap default

swap lvol

swap *hardwarePath*

Adding secondary swap

- Device and file system (dynamic) swap:
The swapon command, usually with the cooperation of the /etc/fstab file

Notes

- **To remove secondary swap:**
 1. Remove it from /etc/fstab
 2. Reboot
- **To move primary swap:**
 1. Change system file if necessary
 2. Run `lvlboot -s` if necessary
 3. Change /etc/fstab
 4. Reboot
- SAM can manipulate swap. It can also add, remove, and modify dump devices too. (Remember `lvrmboot -d` will remove dump devices, and so will rebuilding the kernel.)

Steps to add a non-HP driver

1. Gen the driver into the kernel, add the other files (master.d/*, space.h.d/*) and reboot
2. Run `ioscan -f` or `ioscan -k` and look for the hardware path for a device of class unknown
3. Run `ioscan` again to hotwire the hardware path and the driver:
`ioscan -H 56/16 -M scope`
4. Create the `/dev/` entry with `mksf` (as in `mksf -H 56/16 -d scope`, if the driver returns codes to `mksf`) or `lsdev` and `mknod` (if not)
5. Verify with `ioscan`'s options `-fkn` or `-fun`

Notes

- Adding HP drivers has already been covered.
- Much greater detail is in *Configuring HP-UX for Peripherals*:
 - Appendix C describes the steps to gen and configure a third-party driver, and the format of minor numbers, in the section "Configuring a Device Driver Forcibly"
 - Appendix A tells how to add EISA drivers and other software (by compiling (!) and by creating the `/usr/conf/master.d/*` device table named `$DEVICE`) in the section "Adding Non-HP Drivers to the Kernel Using HP-UX Commands"

Summary

- Single command and file set for kernel generation
- Obsolete programs and arguments
 - S800 uxgen — use mk_kernel
 - The -a option to config — use insf, mksf
- Obsolete filesets and files
 - S800 file and syntax
 - Plain file /etc/master
 - MIRROR for DataPair — use LVM-MIRROR
 - NET, NETINET, NETIPC — use NET-KRN, NET-RUN, NET-PRG, NET-MAN

Notes

- The S800-style drivers will move into maintenance mode. The S700-style I/O is the driver environment of choice.
- Some major numbers are allocated dynamically. STREAMS drivers used to use major numbers from 136 to 145, so suppliers of STREAMS modules/drivers must convert them by:
 - Adding the driver to the master file table
 - Adding the driver table as a separate file in the master directory
 - Adding a small install routine to the driver
 - Using I/O utilities like lsdev in their SD-UX control scripts to find out the proper major number assigned by the I/O system

-
1. Here are some excerpts from four different system files, all are incorrect or not recommended. What's wrong with each of them?

```
swap default
dump 52.3.0
dump default
```

```
* swap default
dump default
```

2. Why would someone want to configure the dynamic buffer cache?

-
1. Run `more` on `/usr/lbin/sysadm/system_prep`. What are two ways to get more output when it runs? _____

- 1.

- 2.

2. Simulate a forgetful customer. What error messages happen when you do the following?

Use a pound sign (#) as a comment character:

Give `mk_kernel` a system file of zero bytes:

Put a control character in the file (try `^L`, `^Z`, `^A` — in `vi`, you will need to type `^V` first):

Misspell or mangle the name of a driver and a tunable parameter:

Build the kernel as a non-super-user:

3. Now build a new kernel, commenting out the `CentIf` driver. How much smaller did the kernel get? (Hint: see file size with `ls -l`, and internal text + data + bss with `size vmunix_test ../vmunix`.) Install and boot the new kernel.

4. Log in and type the following commands:

```
# PATH=/opt/deltaclass/bin:$PATH
# more /opt/deltaclass/bin/feather
```

[continued on the next page]

4. [continued]

```
# time feather; time rm -rf /home/etc
```

Do this last command line twice, and record the times on this page. What's the average run time of the feather command? The rm command?

5. In the system file, set the parameter `fs_async` to 2:

```
* fs_async controls when the file system allocates
* inodes, data blocks, directories, and so on;
* 0 is the default value.
* 0==some async, 1==more async, 2==really async
fs_async      2
```

Then build a new kernel, install it, and reboot. Log back in and type the same commands as above. Figure out the average of the new run times. Is asynchronous allocation faster? How much?

Bonus question: What's the disadvantage of very asynchronous allocation? Think about boundary conditions like file system overflow, error reporting, and problems that can arise if the system goes down.

5. On the Series 700, type the following commands:

```
# PATH=/opt/deltaclass/bin:$PATH
# cd /home
# makelinks links
```

This sets up about 500 symbolic links in a sub-directory called `/home/links/`. Then type the command:

```
# time accesslinks links
```

This script accesses all those symlinks. How long does it take? Run it twice, then write the average of the times on this page.

6. Turn on fast symlinks in your system file and build a new kernel. Did the new kernel get any larger? Install the new kernel and reboot.
7. Now log back in and run makelinks again, this time on a new directory. (Only new links are faster, not old ones.) Time the accesslinks command twice on this new directory. Did access get faster? How much? What percentage is that?
8. *Bonus question:* If you saved the earlier symlinks directory, time a run of accesslinks on that. Do an `ls -ld` on both directories, then do a `du -s` on both directories. What differences are there?

This page intentionally left blank.

Diagnostics



Module goals:

The purpose of this module is to ensure that the student knows the changes to diagnostics, both offline and online, and understands the benefits of Memory Page Deallocation.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Understand the different media that carry the diagnostics.
2. List new and obsolete diagnostics at 10.0.
3. Choose appropriate offline and online diagnostics.
4. Customize a system to start or avoid the diagnostics.
5. Explain how memory page deallocation saves downtime and money.

Motivations for changes

- **There's one HP-UX, there ought to be one Support Strategy**
- **Make diagnostics clearer and easier to use**
- **Support new hardware more easily**
- **Replace diagnostics that have become unmaintainable**
- **Save money on memory boards stocked in field offices**

Notes

Features and benefits

Features

- Same UI on both S700 and S800
 - New interface for Offline Diags (ODE)
- Support Media replaces Supp. Tape/Recovery Sys.
 - On-line Diags no longer on HP-UX Core
- Diagnostics more modularized
- Memory Page Deallocation (MPD)

Benefits

- Easier to use
- Open Architecture
- Lower maintenance costs for lab
- Will save \$s in memory stock costs



Notes

- Because the online diagnostics are on a separate media, customers will only have to install them if they desire them. You no longer need to leave the system undisturbed for a half-hour after an installation in order to load the online diagnostics. Note that the Support Media is part of the HP-UX 10.0 Media Package, and will be included on Instantly Ignited systems as well.
- Memory Page Deallocation (MPD) is now supported in H/W on some models and in S/W on the rest. Regardless, it allows a system to deallocate failing memory on a much more fine-grained level than in the past. This should reduce the need for memory board replacement, and save money on both T&E spent replacing boards and field replacement stocks.

HP-UX 9.0x vs...

- Two on-line diagnostic “Platforms”
 - Sherlock on S800 (sysdiag or DUI)
 - SupportWave on S700 (xstm, mstm, cstm)
- Three types of test programs
 - Diagnostics
 - Exercisers
 - Verifiers
- 3-level licensing for test programs on S800 only
 - Free (programs that merely display info)
 - Licensed (most diags, exercisers, verifiers)
 - HP only (SS_CONFIG)

Notes

- In the past there were two diagnostic environments, one for the S800 and one for the S700
 - Sherlock is designed for CEs or other sophisticated users. It has diagnostics and “utilities” (tools that showed information, like logs and ASCII maps of the hardware).
 - SupportWave is designed for users, and thus is friendlier. It has exercisers and verifiers. It accesses the diagnostics by invisibly calling Sherlock.
- Types of licensing are:
 - **F** — Free
 - **L** — Licensed
 - **H** — HP-only

Note: Licensing permission is granted by the suplicen command.

...HP-UX 10.0

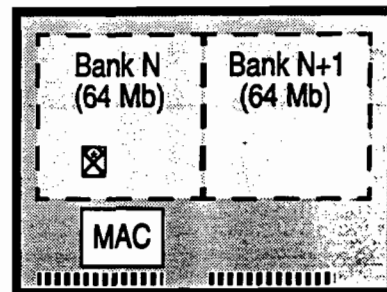
- All 9.0x features now on *both* S700 and S800
 - sysdiag, cstm, mstm, xstm
 - licensing
- Support Media replaces Support Tape (S800) and Recovery System (S700)
- New Offline Diagnostics Environment (ODE)
 - Access via ISL when booted from Support Media
 - You can also run an ODE tool directly from ISL
- Memory Page Deallocation (MPD) introduced

Notes

- There is no longer a S800 Support Tape, nor is the mkrs command provided on either platform. They have been obsoleted.
- SupportWave tools (cstm, mstm, xstm) are now available on both platforms.
- There is a new Offline Diagnostic Environment (ODE) on both machines that provides a standard interface to all diagnostics. It is a modular environment that is intended to both provide a standard interface for diagnostic modules and make it easier to add new ones.

Memory Page Deallocation

- Allows disabling of failing memory at page level
- MPD implemented in S/W or H/W
 - All systems run MEMLOGP if installed
 - On systems with S/W MPD, MEMLOGP keeps disk-based log, and tells MC to remap bad pages
 - On systems with H/W MPD, list is kept in NVRAM



Notes

- Before MPD unrecoverable memory errors required an on-site service call or parts replacement. Some machines (i.e., Emerald) deallocated the bank until the board was replaced.
- Note that the MEMLOGP daemon is part of the on-line diagnostics, and will only be enabled if these are installed.
- With MPD, failed memory is disabled at the page level of granularity
 - HP-UX 10.0 systems use a daemon (MEMLOGP) to look for errors. On systems with S/W-only MPD, the system Memory Controller is notified that there is some bad RAM by the daemon. The list of bad pages is stored in a log file so that it can be reestablished after booting the system.
 - On systems with hardware-based (or "full") MPD, the list of bad pages, called the Page Deallocation Table (PDT) is stored in NVRAM. Full MPD is only on TNT Emerald, Chimera Emerald, and the K Series (Kittyhawk) after 10.0. (Note: The J and K Series machines are scheduled for 1Q95 release.)

Spotting failed memory

- Messages will appear on console
 - Exact content will vary depending on system
- Offline Tools (full-MPD systems only)
 - Use PDC Service Module
`SERVICE> showpdt`
- Online Tools
 - On S/W-only MPD systems
`DUI> logtool`
`LOGTOOL> memrpt`
 - On full-MPD systems
`DUI> memdiag`

Notes

- On a full-MPD machine, replacing the memory array or changing the memory configuration will erase the PDT, which will be rebuilt over time. This means that there's little if anything to be gained from swapping cards around in the machine. Also, it means that rebuilding of the PDT is automatic.
- On a S/W-based MPD system, the on-disk log file needs to be explicitly cleared when memory cards are replaced or moved. Here's the procedure:
 1. # `sysdiag`
`DUI> logtool`
`LOGTOOL> memclr`
 2. # `shutdown -h gracePeriod`
 3. Replace the board or change the configuration.
 4. Reboot.

Support media at 10.0

Name	Contents	CD	DDS	1/2"	1/4"	QIC
Support Media	LIF/ISL, all diagnostics (offline + recovery, online + RCL) for S700 & S800	700 800	700 800	800	800	800
Service Tools	LIF/ISL + SS_CONFIG only (no Online Diags)	700 800	700 800	800	800	800
Instant Ignition	Online diagnostics	—	—	—	—	—

- The customer must install online diagnostics and RCL tools (HP Predictive) from the SD-UX depot on the Support Media

Notes

- The Support Media contains an SD depot with the Online Diagnostics. Use this to load Online Diagnostics onto a system following installation.
- The Support Media is a bootable image that contains the Offline Diagnostics and also serves as a recovery environment. Note that while the S700 and S800 Support Media have different part numbers, they have the same contents and are interchangeable.
- A CE needing SS_CONFIG will still need a separate bootable media (the Service Tools) containing this tool. Note that the Service Tools media does not contain the Online Diagnostics.
- The CTD 1/4" solution will have less functionality than the CD or DDS. After 10.0, Support Media on CTD will be obsoleted.
- Note that the Support Media is part of the HP-UX Media Package, so most customers will get it. It is also installed on systems that are Instantly Ignited, so they will have it as well.
- Note that Online Diagnostics are not include in the Desktop HP-UX configuration, and are disabled when a system is converted to "Lite".

Changes to offline diagnostics

Replaced	Replacement or addition
Sherlock disk utils	DISKUTIL for novices
Sherlock disk utils	DISKEXPR for experts
DTDUTIL	COPYUTIL disk recovery
	DFDUTIL disk firmware download
EISADIAG	EISATEST tests EISA
IOMAP	IOTEST tests I/O devices
IOMAP	PERFVER peripherals self-test
IOMAP	MAPPER map of I/O system
	PDIAG J & K processor
	U2TEST J & K I/O chip
	WAXTEST J & K wax
	XDIAG for GSC X25 card
	PCVU to fix up stable storage

Notes

- Of course, the ODE shell for running diagnostics is new too. The long-term strategy is to migrate the ISL diagnostics to ODE.
- Obsolete Sherlock utilities include diagnostics and disk utilities:

Diagnostic	Disk utility	10.0 disk utility replacement
CENTPBA	CS80DIAG	DISKUTIL, DISKEXPR
AFIDAD	FLEXDIAG	DISKUTIL, DISKEXPR
GPIODAD	SCSIDSK2	DISKUTIL, DISKEXPR, DFDUTIL
SS80DIAG	SARRAY	DISKUTIL, DISKEXPR, DFDUTIL
FDDIEISA	DTDUTIL	COPYUTIL

Diagnostics are obsolete because either the cards are not supported at 10.0 (CIO AFI, NIO GPIO), or a survey of field and manufacturing revealed that nobody was using the diagnostics (SS80DIAG, CENTPBA).

- PCVU is downloaded to the uxboot1f file when diagnostics are installed from the Support Media.

Changes to online diagnostics

Changed	Added
3480 supported by supplier	Memory Page Deallocation SCSI3480
REWRITE sent to RC as needed	MODMUTIL for integrated K modem
LANDAD for Wright Brothers	REWRITE shipped to customers LANHSC for core LAN in WB and K HARRAY for HA F/W SCSI (Nike) Driver logging for GSC & F/W SCSI
SX exerciser	SupportWave enhancements: - CPU exercisers - X25DAF for GSC X25 card - TSDAF for J-Series GSC Teleshare - SCSI ISDN diagnostic - GRAPHDAF for J & K Series

Notes

- The biggest impact is with MPD.
- Remember that the 10.0 diagnostics are not installed at upgrade or install time. The diagnostics are shipped on the Support Media, and the customer must install them as a separate step.
- The 9.0 diagnostics will not run on 10.0. At upgrade time, all 9.0 diagnostic binaries are removed from the system.
- SX system exercisers will move to SupportWave ([xmc] stm).
 - Customers must rewrite any SX scripts in cstm commands
 - Exercisers are licensed; run suplicen before the *stm program
 - No online help; see the manuals for error-message explanations
- Drivers for GSC SCSI and F/W SCSI do logging now for SupportWatch and field troubleshooting.
- The disked command is no longer shipped (as it was on the 9.0x S800 Support Tape), and has never been supported.

Tips and tricks

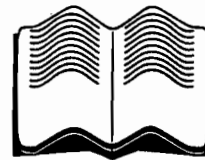
- Diagnostic daemons use IPC, be careful when tuning IPC parameters and killing daemons
- Recovery process detailed in *Support Media User's Guide*

Notes

- Diagnostics still use inter-process communication (IPC); do not try to stop them with `kill -9` because it will not free up system memory.
 - SX scripts will need to be rewritten.
 - Recovery is not fully sketched out at this point, but the steps will be described in the *Support Media User's Guide* at release:
1. Boot from Support Media.
 2. Troubleshoot and repair the disk.
 3. Format the disk.
 4. Lay down a generic LIF volume.
 5. Build `/sbin` and copy in some archive (non-shared) commands, including the recovery tools and a temporary kernel.
 6. Boot from the repaired disk.
 7. Using your `frecover`, `tar`, or `cpio` backups, restore data to the lost disk(s). Only these three backup formats will be supported.

For more information

- *Support Media User's Guide*
- *Memory Page Deallocation Familiarization Guide (CE42-MEMPAGE)*



Notes

-
1. The three levels of licensing have not changed. So what's new about licensing at 10.0?
 2. When *must* you use an offline diagnostic instead of an online one? Why can't you use the Service Tools tape anymore to get the offline diagnostics?
 3. Where are we keeping the diagnostics nowadays? How do you get them for new systems? For old upgraded ones?
 4. How does memory page deallocation work across reboots? How does it save money for the customers?
 5. What are the hardware requirements for running xstm? For mstm? For cstm?

Your instructor will provide you with diagnostics tapes for this exercise.

- 1. What kind of programs can you run under ODE?
Record your commands below or in the margins.**

Boot up the offline diagnostics tape and bring up ISL:

Run ODE:

Find out what's in ODE:

- 2. Exit ODE, then run the ODE tool named MAPPER from the ISL prompt. After ODE runs MAPPER, does it stay around or does it exit?**
- 3. Run MAPPER from inside ODE. Get help on it. Run the diagnostic part of it.**
- 4. Run PERFVER. Which section tests SCSI devices? Run only that section.**
- 5. Run DISKUTIL. Set the loop count to 2.**

Which sections are run by default?

Run only sections 1 and 2 together.

Run only section 3, but interrupt it and terminate the test early. Get back to the DISKUTIL> prompt.

Set the loop count back to 1.

6. Run DISKEXPT. Get help on it. What can it do, specifically?

7. Run COPYUTIL. What version do you have? Get help to find out how to run it, as if you were going to restore a system from tape. In fact, you will do exactly that on Friday — more on that below.

8. Run LASIDIAG. Which sections test the keyboard?

Run those sections.

Which section tests the workstation powerfail test?

Run that test and reboot your system.

9. If you have extra time and want to get some experience with the online diagnostics, install them on your system. Your instructor will indicate the location of a depot containing the software.

10. Return your diagnostics tape to the instructor.

This page intentionally left blank.

SAM and Administrative Commands



Module goals:

The purpose of this module is to enable the student to use new features of SAM to maintain the system, and to understand how other administrative commands have changed.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe changes to and use SAM's modified functional areas.
2. Describe changes to SAM's logging, and use the log file viewer.
3. Use Restricted SAM Builder.
4. Add, modify and remove custom SAM applications and application groups.
5. Use new and modified administrative commands.

Motivations for changing

- **Enable SAM and the administrative commands to support new standards and features of HP-UX**
- **Standards:**
 - **SVID3**
 - **SVR4**
 - **XPG4**
 - **POSIX.2**
 - **4 byte EUC**
- **New features supported include:**
 - **LVM on S700**
 - **Kernel configurations**
 - **I/O convergence**



Notes

- **When available, JFS and NFS Diskless will also be supported by the system administration commands and utilities, including SAM.**

Features and benefits

Features

- Administrative commands standards conformance
- Non-root user access to limited administrative tasks
- SAM's enhanced user interface

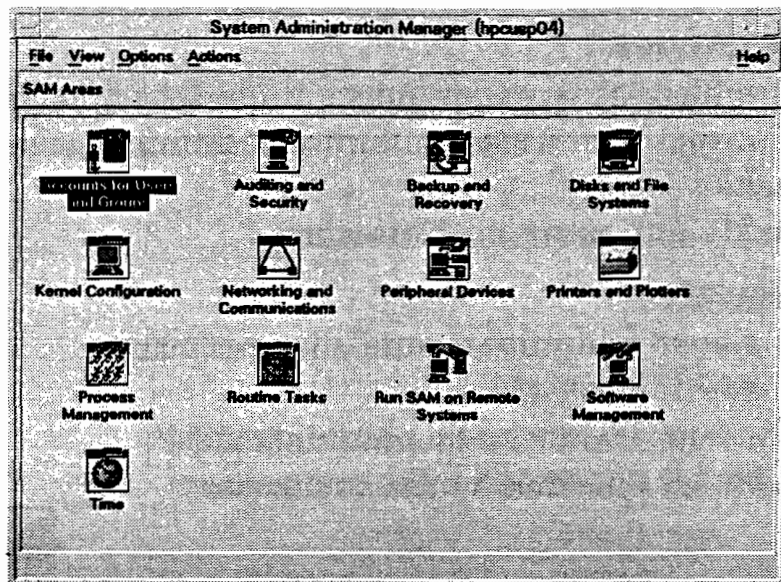
Benefits

- Easier to administer multiple platforms
- Easier to use SAM
- Non-root users can do administrative tasks, as specified by the superuser



Notes

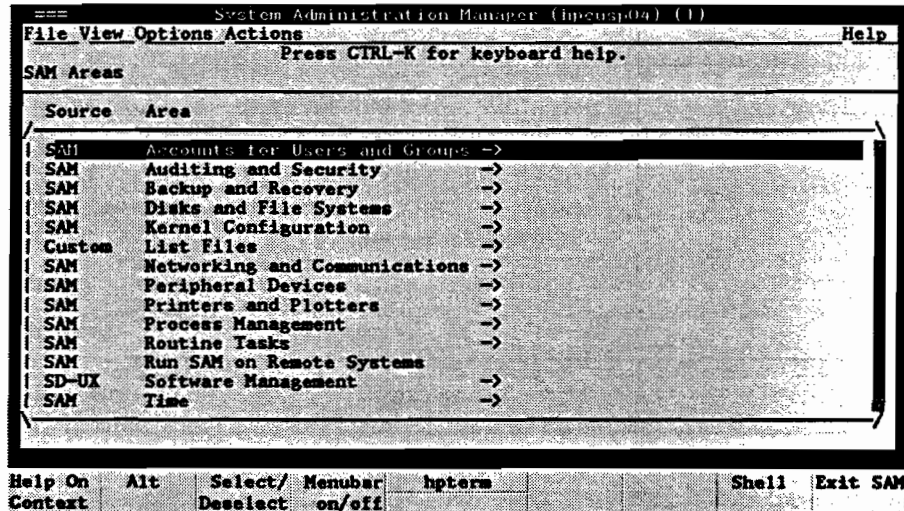
SAM FAL: X11/Motif interface



Notes

- The SAM Functional Area Launcher (FAL) replaces the 9.0x Control Box.
 - The 9.0x Control Box and the SAM FAL are the first screen displayed in their respective SAMs.
- Provides new menu options:
 - *File* to print object-list data (to file or printer) or exit SAM
- New options to *Options* and *Actions* are covered later in this module
- There are two interfaces supported by FAL
 - X11/Motif
 - Character terminal
- The X11/Motif interface provides iconic or tabular FAL.
- Can move between the two by selecting:
 - *View...By Name and Icon* for iconic view
 - *View...By Properties* for tabular listing view
- To modify SAM's appearance if using Motif interface, edit file:
/usr/sam/lib/X11/app-defaults, per system's X Windows GUI doc.

SAM FAL: ASCII interface



System Administration Manager (hncusp0a) (1)

File View Options Actions Help

Press CTRL-K for keyboard help.

SAM Areas

Source	Area
SAM	Accounts for Users and Groups ->
SAM	Auditing and Security ->
SAM	Backup and Recovery ->
SAM	Disks and File Systems ->
SAM	Kernel Configuration ->
Custom	List Files ->
SAM	Networking and Communications ->
SAM	Peripheral Devices ->
SAM	Printers and Plotters ->
SAM	Process Management ->
SAM	Routine Tasks ->
SAM	Run SAM on Remote Systems ->
SD-UX	Software Management ->
SAM	Time ->

Help On Alt Select/ Menubar hpterm Shell Exit SAM
Context Deselect on/off

Notes

- Lists functional areas in alphabetic order, identifying source of each.
 - Actually, all interfaces display source, including icons, if specified.
- *SAM* for SAM functional areas
- *Custom* for customer-added applications
 - Adding these is covered later in this module
- *Other* for non-specified HP or third-party applications
 - HP or third-party applications can specify own source
 - Example: SD-UX added itself to SAM, and specified source as "SD"
- NOTE: Third-party applications added to SAM's FAL are supported by the third party, not by HP
- Double-clicking on any object in an object list invokes a "zoom" or other similar action, providing more detail (usually) on the selected object.
 - In FAL, double-click executes application or opens application group.

"Password Options" screen

The screenshot shows a dialog box titled "Password Options" for the user "samuel". It contains a list of password options: "No Restrictions (Normal Behavior)", "Force Password Change at Next Login", "Allow Only Super-User To Change Password", and "Enable Password Aging". The first option is selected. At the bottom are "OK", "Cancel", and "Help" buttons.

Login Name: samuel
Password Options
No Restrictions (Normal Behavior)
Force Password Change at Next Login
Allow Only Super-User To Change Password
Enable Password Aging
OK Cancel Help

Notes

- **Users changes**
 - Add user to or remove user from multiple groups in one dialog box
 - Optional password modifications, as shown above
 - Password aging allows administrator to specify minimum and maximum weeks for which a password can exist before changing
- **Groups changes**
 - Can specify group ID when adding a group
 - Can change group name and/or group ID when modifying a group
- For trusted systems, additional options automatically come up.
 - See on-line help for this area for details.
 - Additionally, trusted systems features are described in "Trusted Systems" module.

**"Backup and Recovery"
changes**



"Select Backup Scope" screen

Select Backup Scope (Upgsyspl)

Backup Scope: **Specified Files**

Cross NFS Mounts: **No**

Type in the files or directories to be included and excluded.

Included Files		
Name	Type	Location
/home/samuel	NFS	/dev/vg00/lvol14
/etc	BFS	/dev/vg00/lvol1

File Name: /etc

Add Modify Remove

Excluded Files (optional)		
Name	Type	Location
/home/samuel/.X11-unix	NFS	/dev/vg00/lvol14
/home/samuel/iso	NFS	/dev/vg00/lvol1

File Name: /home/samuel/junk

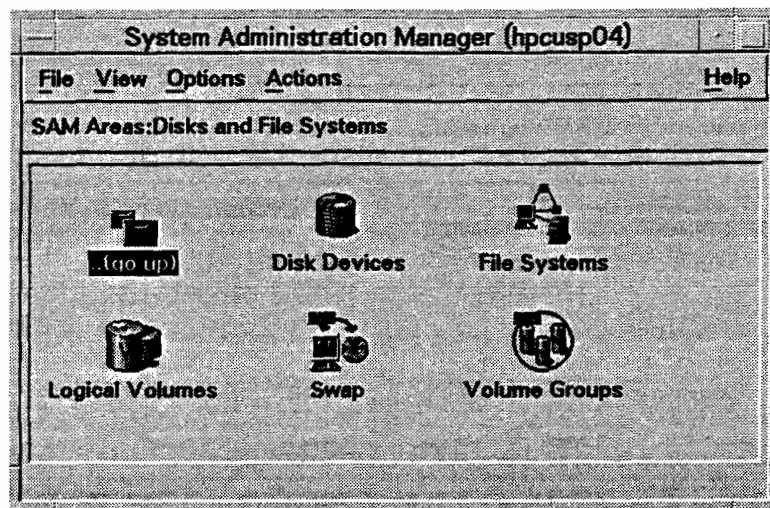
Add Modify Remove

OK Cancel Help

Notes

- This is now divided into two subareas.
 - Setting up and managing automated backups
 - Interactive backup and recovery
- On-line help has been rewritten completely for this functional area.
- Backup scope selection screen has changed.
 - More selection alternatives
 - Displays file system type and device location of selected files
 - Simplifies NFS mount handling
- Backup device selection screen has changed.
 - Displays device characteristics list for selection
 - Can create custom device files

"Disks and File Systems" screen



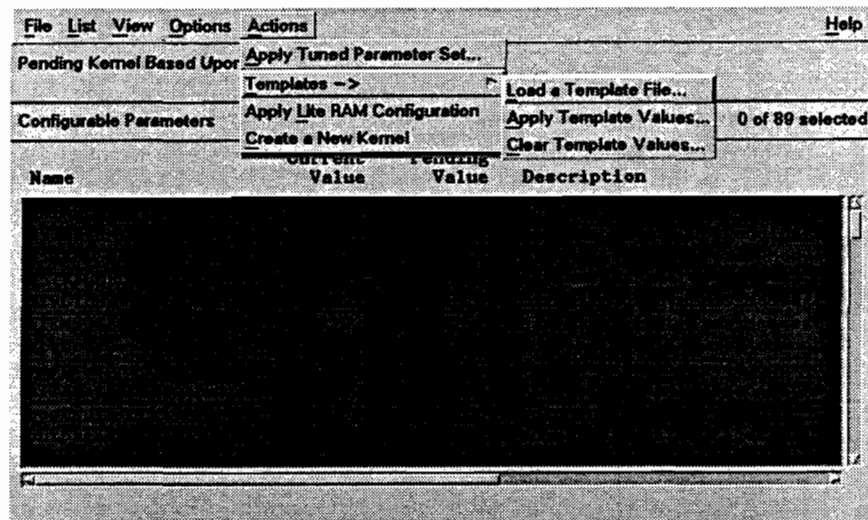
Notes

- *Disk Devices* manages all devices from one screen.
 - Includes magneto-optical drives and autochangers now
 - Shows all disks recognized by ioscan/diskinfo
- *File Systems* manages all file system types from one screen.
 - NFS and local
 - Many more configuration options for mounting remote file systems
 - Can disable and enable automounter
- *Logical Volumes* supports S700 and S800.
 - Accesses all LVM commands
 - Can create desired file system or set swap when creating logical volume
- *Swap* modifies primary swap.
- *Volume Groups* can create, configure, name and extend volume groups.

"Kernel Configuration" changes



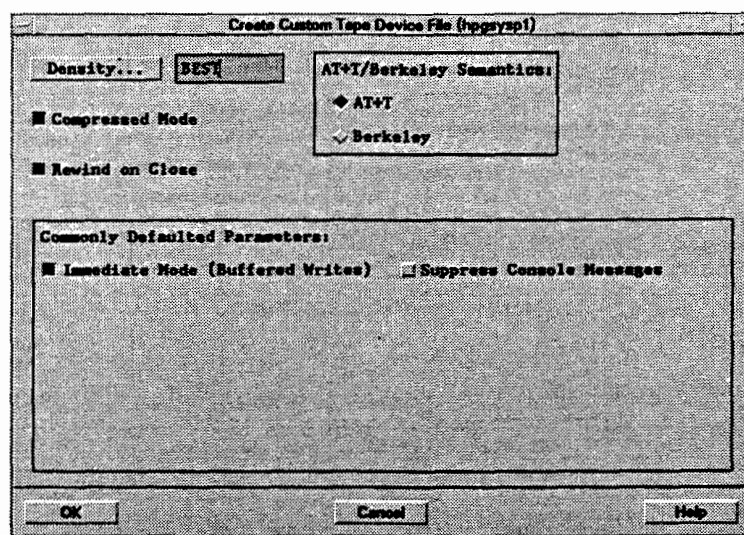
"Kernel Configuration" changes



Notes

- Tunable parameters and kernel build process are converged.
- Can no longer configure root and console devices in kernel.
- Displays optional CDIOs on drivers list.
- The Dump device configuration screen is new subarea.
- Can apply quick configurations
 - Lite RAM for S700 with <32 MB RAM, reduces kernel space
 - Tunable parameter sets

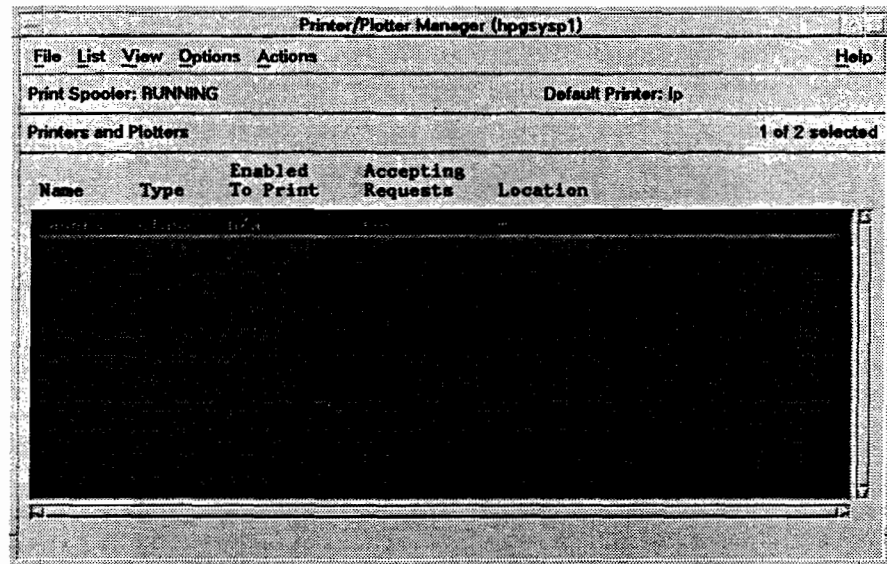
"Custom Tape Device" screen



Notes

- Allows custom tape device file creation.
 - The mksf command is used to create device file.
- The above dialog box is for a tape2 device file.
 - The tape1 and tape2 drivers allow for suppression of console messages.
 - The stape driver allows exhaustive mode, partition control and variable or fixed block sizes.
 - Other characteristics displayed are available for all tape device files.
- Default device files are created by executing insf.
- Device files are recognized and created using I/O convergence definitions for device file names and minor number formats.
- Refer to the "I/O Convergence" module for a review of the I/O convergence definition for device file names and minor number formats.
- Other major changes for this area are in terminals and modems subarea, but for trusted systems only. Refer to on-line help for this area.
 - Additionally, trusted systems are covered in "Trusted Systems" module.

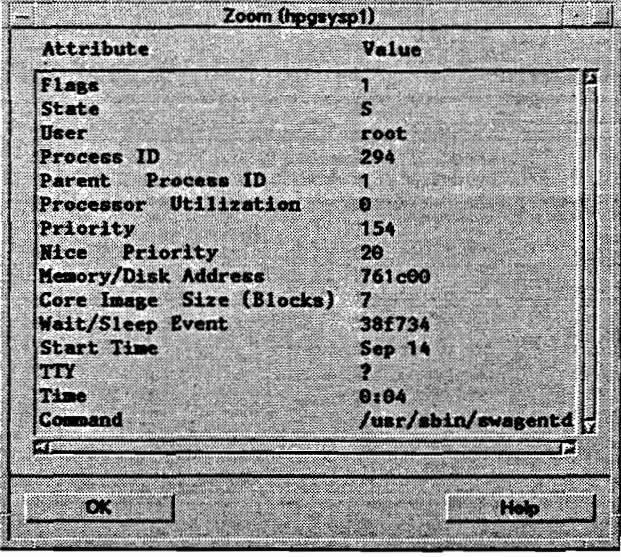
"Printers/Plotters" screen



Notes

- Object-list screen display
 - "Status" split into two columns:
 - "Enabled to Print"*
 - "Accepting Requests"*
 - *"Fence Priority"* column not shown by default
Use View menu to see
- Shows printer/plotter and print classes on which actions can be performed.
- New actions can be performed without enabling or disabling printer:
 - *Accept Requests for Printer / Plotter / Class*
 - *Reject Requests for Printer / Plotter / Class*
- Selection of multiple objects permissible for most actions.
- There are no restrictions on which text characters can be used for a remote printer.
 - Provides better interoperability with non-HP-UX systems

"Process Management"--Zoom



The screenshot shows a window titled "Zoom (hpgsysp1)" with a table of process attributes. The table has two columns: "Attribute" and "Value". The attributes listed are: Flags (1), State (S), User (root), Process ID (294), Parent Process ID (1), Processor Utilization (0), Priority (154), Nice Priority (20), Memory/Disk Address (761c00), Core Image Size (Blocks) (7), Wait/Sleep Event (38f734), Start Time (Sep 14), TTY (?), Time (0:04), and Command (/usr/sbin/swagentd). At the bottom of the window are "OK" and "Help" buttons.

Attribute	Value
Flags	1
State	S
User	root
Process ID	294
Parent Process ID	1
Processor Utilization	0
Priority	154
Nice Priority	20
Memory/Disk Address	761c00
Core Image Size (Blocks)	7
Wait/Sleep Event	38f734
Start Time	Sep 14
TTY	?
Time	0:04
Command	/usr/sbin/swagentd

Notes

- More process attributes available for display.
 - *Actions...Zoom* displays all attributes for selected process
 - Or double-click process
- Automatic refresh interval can be specified for "*Process Control*" list.
 - 1, 5, 10, or 30 seconds
 - 1 minute
- There are more performance monitors available in the "*Performance Monitors*" subarea.

"Routine Tasks" screen—files

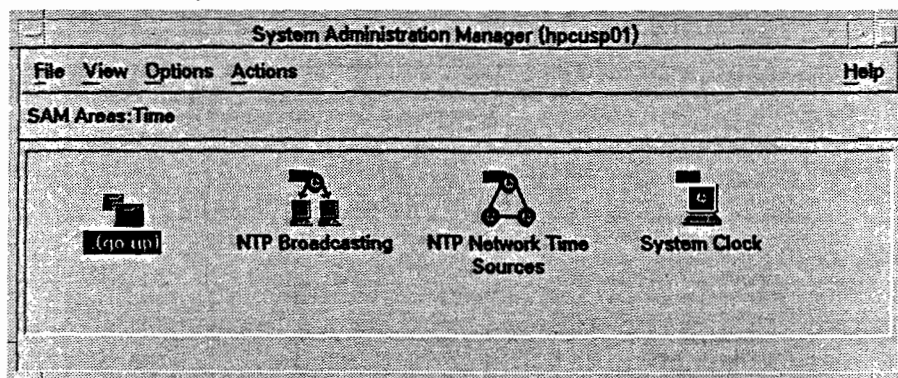
Routine Tasks (hpcusp04)			
File	List	View	Options Actions Help
System Log Files		0 of 49 selected	
File Name	Percent Full	Current Size (Bytes)	Recommended Size (Bytes)
/etc/aisa/config.err	NA	NA	1024
/etc/aisa/config.log	NA	NA	1024
/etc/rc.log	102	10490	10240
/etc/rc.log.old	NA	NA	10240
/etc/shutdownlog	0	0	1024
/var/adm/OLMessages	NA	NA	4096
/var/adm/OLDrid.log	NA	NA	1024
/var/adm/OLDsolog	NA	NA	4096
/var/adm/backuplog	NA	NA	1024
/var/adm/btmp	NA	100	NA
/var/adm/cron/OLDlog	NA	NA	32768
/var/adm/cron/log	0	170	32768

Notes

- Selective file removal
 - Unowned files can also be specified
- Log file tailed for viewing
- System log file list displays every file that SAM looks for, whether or not it exists on system.
 - ASCII and non-ASCII on same list
 - New column indicates whether or not file exists on the system.
 - Users can add to this list.



"Time" screen



Notes

- This is a new functional area.
- User can set the system date and time.
- For NTP (Network Time Protocol), user can:
 - configure NTP daemon
 - specify local time source
 - define time source authentication policy
 - add, modify and remove NTP server(s)
 - configure server to send out regular time broadcasts to all hosts
 - configure NTP local clock

Overview of other major changes

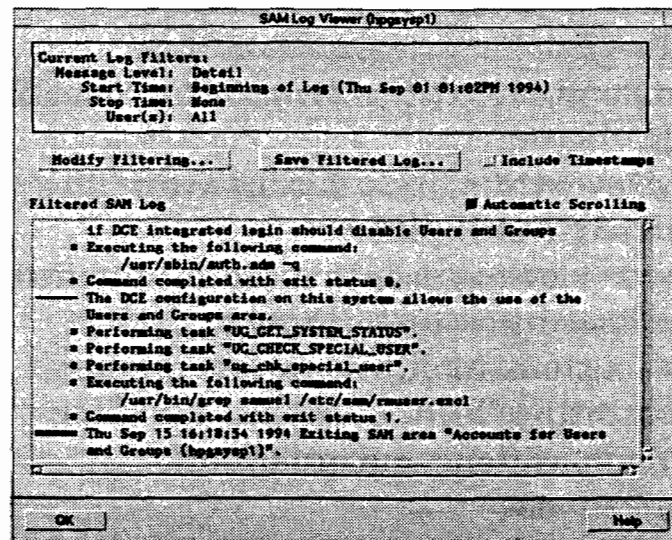
- On-line help extensively updated
- Includes list of new features on-line
 - Menu option *Actions...What's New in SAM*
- Log file viewer
 - Logs everything now, but user can filter view
- Restricted SAM Builder
 - Allows non-root user to perform specified administration tasks through SAM
- Adding custom applications and groups
 - From within SAM, users can add their own applications and groups of applications



Notes

- The log file viewer, Restricted SAM Builder and adding custom applications and groups are described in more detail on the next pages.

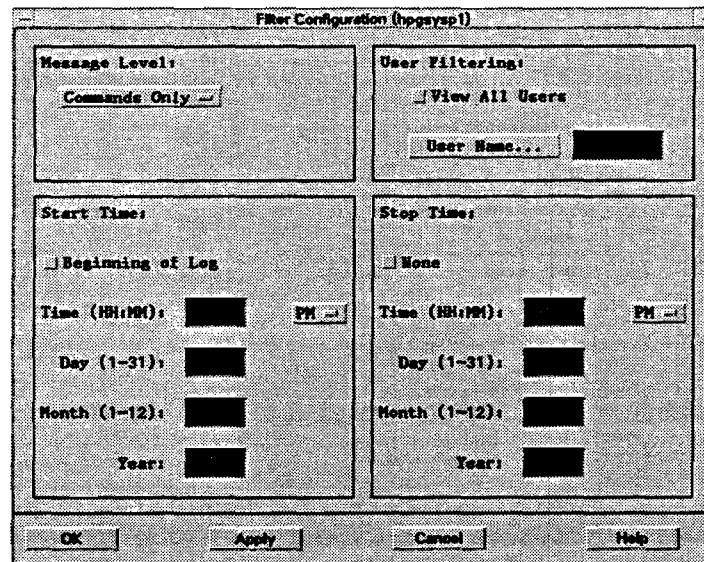
SAM Log Viewer



Notes

- Use SAM Log Viewer from within SAM, or from command line.
 - *Options...View SAM Log* or `/usr/sam/bin/samlog_viewer`
- Can run multiple SAM Log Viewers simultaneously.
- All messages always logged.
 - Users no longer specify logging level to store in log file, but they can specify which logging level they would like to view.
 - Use "Filter Configuration" dialog box to modify view.
- Log file name cannot be changed, but can use symbolic links.
- Log file not formatted for easy viewing with common text display commands (e.g., vi, more).
 - Has additional characters, so use SAM Log Viewer instead.
- Use *Options...View Log Options* to
 - Specify any maximum size for log file. (even "0" saves last session)
 - Never automatically trim log file (trimming saves one trim to samlog.old)
 - Auto-start SAM Log Viewer.

Log file viewer filtering options



Notes

- Access through SAM Log Viewer's *Modify Filtering...* button.
- Can specify what level of message to view.
 - *Commands Only* is new level
- Can specify to view events performed by all users or a particular user only.
 - Useful for Restricted SAM Builder, which is discussed on next page
- Can specify to view log from start date and time, to stop date and time.
- Output filtered log to an ASCII text file, using SAM Log Viewer's *Save Filtered Log...* button.

Restricted SAM Builder screen

File	View	Options	Actions	Help
Privileges for: samuel			Open Item	Changes Pending: YES
SAM Areas			Close Level	
			Go to Top Level	1 of 15 selected
Source	Area	Save User Privileges...	Access Status	
SAM	Account	Remove User Privileges...	Super ->	Disabled
SAM	Audit	Load User Privileges...	->	Disabled
SAM	Backup	Enable All	->	Enabled
SAM	Cluster	Enable	->	Disabled
SAM	Disks	Disable	->	Partial
SAM	Kernel	Disable	->	Disabled
Custom	List Files		->	Disabled
SAM	Networking and Communications		->	Disabled
SAM	Peripheral Devices		->	Partial
SAM	Printers and Plotters		->	Enabled
SAM	Process Management		->	Partial
SAM	Routine Tasks		->	Enabled
SAM	Run SAM on Remote Systems			Inaccessible

Notes

- Build a SAM for non-root users to access partial SAM functionality.
 - Actually, a setuid root executable verifies user has access.
 - If so, user's restricted SAM is run.
 - If not, error message indicates user must be super-user to run SAM.
- "Access Status" column indicates default access to item for non-root users.
 - "Enabled": non-root users can access all options of this item
 - "Disabled": non-root users cannot access any options of this item
 - "Partial": some options in this area not available to non-root users
 - "Inaccessible": not available to non-root users, cannot override
- Default access status can be overridden when building restricted SAM
 - "Inaccessible" status is exception--cannot be overridden

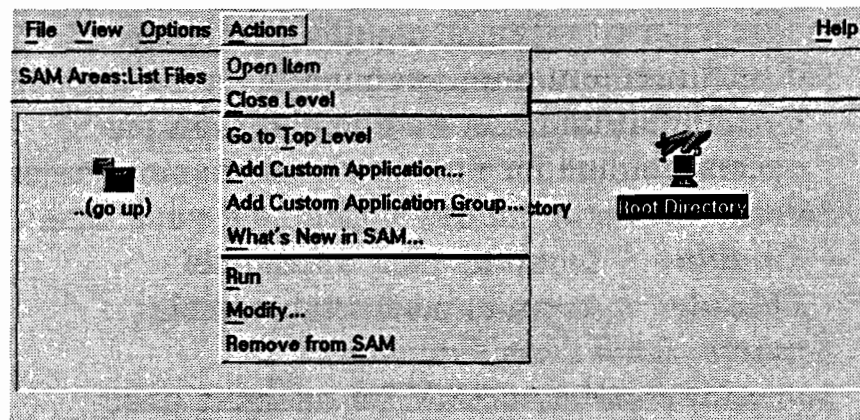
Using Restricted SAM Builder

1. Execute: `/usr/sbin/sam -r`
2. Select user to set up or modify privileges.
3. Select functional areas to modify access status.
 - SAM defaults areas as disabled that are recommended for non-root users to not access
4. Select *Actions* to modify options for each area
 - *Enable* to make user-accessible
 - *Disable* to make user-inaccessible
5. *Actions...Save User Privileges*
to create restricted SAM for selected user.
6. User executes: `/usr/sbin/sam`
to run personal version of restricted SAM.

Notes

- Must be super-user to run Restricted SAM Builder.
- Build SAMs for individual users, not groups.
 - Use existing logins or a special administrative login
 - For example, have a user called "backup" or "printadm"

Example of custom applications

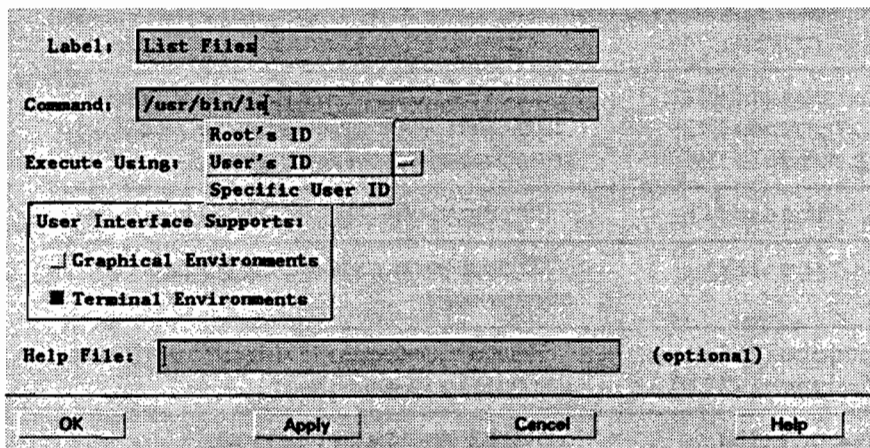


Notes

- The above picture shows custom applications within a custom group.
 - Icons provided by SAM display for all custom applications and groups.
- A custom application has been selected, and *Actions...* submenu has been selected, to show options available for selected custom items.
 - *Run* executes application (same action as double-clicking item).
 - *Modify* displays "Add Custom Application" dialog box (or "Add Custom Application Group" dialog box, if custom group selected).
 - *Remove from SAM* removes item from SAM FAL, but not from system.
- To create custom group:
 1. From SAM FAL menu, select: *Actions...Add Custom Group*
 2. Complete dialog box:

<i>Label</i>	group name as displayed in SAM FAL
<i>Help File</i>	absolute pathname for custom group's help file, to be displayed when user selects group and presses <F1>

“Add Custom Application” window



Notes

- To add a custom application, follow these steps:
 1. Be at the level (group or top) to which you want to add application.
 2. From SAM FAL menu, select: *Actions...Add Custom Application*
 3. Complete dialog box:

<i>Label</i>	application name as displayed in SAM FAL
<i>Command</i>	full pathname of command, include appropriate options
<i>Execute Using:</i>	execute as root, user who started SAM, or specific user
<i>UI Supports:</i>	Whether custom application will execute with or without hpterm when run in Motif environment Can select both
<i>Help File</i>	location of custom application's help file, displayed when user selects <F1> when running application
 4. Test it out!

New administrative commands

Command	Remarks
<code>fstyp (1M)</code>	Determine file system type in MFS environment
<code>groupadd (1M)</code> <code>groupmod (1M)</code> <code>groupdel (1M)</code>	SVID3, SVR4: Command-line interface to manage groups—add new groups, delete groups, modify some characteristics of existing groups
<code>listusers (1)</code>	SVID3: displays GECOs info about user-level logins
<code>logins (1M)</code>	SVID3: displays info about selected groups of user or system logins
<code>mountall (1M)</code> <code>umountall (1M)</code>	SVR4: mounts/unmounts all removable file systems listed in the file system table
<code>pwconv (1M)</code>	SVR4: converts to secure passwords
<code>setuname (1M)</code>	Modify system name and/or node name

Notes

- See man pages for details about syntax and usage.
- MFS is abbreviation for “multiple file systems”.
- GECOs refers to the comment area (e.g., where the user’s “real” name, phone number, etc. are included) for each entry in `passwd`
- Non-superusers can run `listusers` command, but only superuser can run `logins`.
- For `mountall` and `umountall`, can limit action to local or remote file systems.
- If commercial security not installed, `pwconv` removes passwords from `/etc/passwd` and creates commercial security database. If it is installed, `pwconv` updates database to accurately reflect condition of `/etc/passwd`.
 - Other security command changes covered in “Trusted Systems” module.
- The `setuname -n` and `uname -S` commands are identical in functionality.
- The `setuname -s` is only supported way to change system name.
 - Changes can be temporary, or permanent (e.g., survive reboots)
 - If permanent, will change `/etc/rc.config.d/netconf`, and system name and/or node name environment variables change.

New administrative commands

Command	Remarks
setup (1M)	similar user interface to SAM, to do some system setup tasks: set date, time, timezone, nodename, add users, assign passwords to system and admin logins that did not possess passwords
sysadm (1M)	set of logins/commands to do limited admin tasks:
checkfsys	check file systems
makefsys	make file systems (all of these but
mountfsys	mount file systems checkfsys access
powerdown	shutdown system SAM functional areas)
umountfsys	unmount file systems
useradd (1M)	SVR4: Command-line interface to manage users—
usermod (1M)	add users, set defaults for new users, delete users,
userdel (1M)	remove their home directory, modify existing users

Notes

- The setup command provides a common approach across multiple UNIX platforms for common system setup tasks.
 - Use with SAM to setup system, or to make changes after system in use.
 - Requires 1 MB of disk space
 - Works in both X11 and terminal-based environment
- The sysadm command and login invokes SAM.
 - The other commands provide specified system administration capabilities, allowing system administrator to delegate these tasks to others.
 - Passwords provide access to these logins.

Multiple file systems support

- Some file system administration commands modified for multiple file system (MFS) support
- Small “wrapper” to determine file system type
 - Executes file system-specific binary to do work
- These commands have wrapper for MFS support:

ff	fsck
fsdb	labelit
mkfs	ncheck
newfs	quot
quotacheck	volcopy

Notes

- Small “wrapper” determines file system type.
 - Indicated on command line using -F
 - Or will use /etc/fstab or /etc/default/fs
- It looks at per-file system configuration files.
 - In /sbin/lib/mfsconfig.d
 - Not meant to be changed.
- It executes per-file system binaries.
 - In /sbin/fs/<FStype> or /usr/sbin/fs/<FStype>
 - Not meant to be executed directly.
- This change to commands should be invisible, other than the syntax changes as noted on the next pages.
 - Exception: if configuration files or per-filesystem binaries are changed or removed, these commands could stop working or behave strangely.
- See new man page, fs_wrapper (5), for details.

Multiple file systems support

- “Wrapped” commands support new MFS options:
 - F <FStype>
 - o <file_system_specific_options>
 - V (verify—usually, echoes command line)
- Exceptions: these do not support new -o option:
 - labelit quot quotacheck
- These commands support all new MFS options:
 - df mount/umount
- These support new MFS options, but for -o:
 - dcopy

Notes

- Currently, only HFS supports dcopy. Error returned from command on other filesystems.
 - This command was supported on S800 only at 9.0x, and is now also supported on S700.
- All wrapped commands now have multiple man pages. One man page is for the generic usage, the others are file system specific.
 - e.g., df_hfs, ff_hfs
- Refer to man pages for each command for additional syntax and usage information, including file system specific options, if available.

Miscellaneous admin commands

Command	Changes
bdf	-t works with MFS; -i displays "-1" as # of free inodes for FS that do not have fixed inode #; output more readable
df	new options report add'l info for inodes, blocks, KBs, files
fuser	new options: -c specifies file is mount point, any usage in FS reported; -f specifies file is treated as regular file, even if FS; output reports process' text file as t, open files as o
mount	New -v option displays FStype in addition to old output; new -e option replaces old -v (backwards compatibility)
ncheck	Separates -i <i>inode_#s</i> with comma, not space/blank

Notes

- Many of these commands have changed to conform to SVID3 standard or SVR4 specifications.
- See man page on individual commands for more information.

Miscellaneous admin commands

Command	Changes
<code>passwd(1)</code>	superuser (only) can set aging parameters
<code>savecore</code>	S700 and S800 have same features—9.0x features of S800, and new features: default runs <code>savecore</code> every reboot if panic and has space, saves corefile to <code>/var/adm/crash</code> , other features available to save specified troubleshooting info
<code>shutdown</code>	brings system to run level 0 before halting, rebooting or placing system in single-user state, replacing actions of 9.0x <code>shutdown</code> and <code>/etc/shutdown.d</code> scripts
<code>syslogd</code>	logs kernel messages through special kernel log device <code>/dev/klog</code> ; new option <code>-D</code> prevents kernel from printing messages on console, log default: <code>/var/adm/syslog/syslog.log</code>

Notes

- Obsolete administrative commands (no longer available on HP-UX 10.0):
 - `mkrs`
No replacement (boot from Support Media)
 - `monitor` (unsupported, but available on 9.0x)
Alternatives include HP Glance/Glance Plus products, and data from such HP-UX commands as `ps`, `top`, `vmstat`, `iostat`, `sar`
 - `uxgen`
Replaced by `config`, as described in “Kernel Generation” and “S700 and S800 I/O Convergence” modules
- Refer to “User Environment” module for modified and new user-level commands.
- Refer to “S700 and S800 I/O Convergence” module for commands that have been affected by I/O convergence.

Changes to newfs command

- Generic usage:

```
newfs [-F <FStype>] [-V]
      [-o <specific_options>] special
```

- HFS-specific usage:

```
newfs [-F hfs] [-V] [-S|-L] [-d] [-v] [-R swap]
      [-B] [-O disk_type] [mkfs_options] special
```

HFS option	Description
-d (disregard)	replaces old -F (force) option, (-F option now for FS type)
-R swap	reserves swap MB of swap/boot space past end of FS
-B	reserves space for boot programs past end of FS
-O disk_type	specifies disk type in /etc/disktab

Notes

- For newfs -B option:
 - Reserves space sufficient for /usr/lib/uxbootlf
 - If that file not present, reserves 691 KB
- By default, newfs does not use /etc/disktab for disk parameters. Uses:
 - FS size: disk size returned by DIOC_CAPACITY (minus swap/boot)
 - block size: 8192
 - fragment size: 1024
 - tracks per cylinder: calculated based on FS size
 - sectors per track: calculated based on FS size
 - rpm: 3600
- Default bytes per inode value increased from 2048 to 6144.
- Obsolete:
 - disk_type runstring parameter
 - -n option (S300/S400 option)

System accounting

9.0x directory/file	10.0 directory/file (replaces)
/usr/adm/acct/	/var/adm/acct/
/usr/lib/acct/	/usr/sbin/acct/
/usr/lib/acct/holidays	/etc/acct/holidays

- Start accounting system at boot by setting `START_ACCT` to 1 in `/etc/rc.config.d/acct`
- For `runacct`, `CONNECT` argument replaces `CONNECT1` and `CONNECT2`
- Four new accounting commands: `closewtmp`, `utmp2wtmp`, `acctcon`, `acctprc`
 - Under normal usage, these are not invoked

Notes

- Conforms to HP-UX 10.0 File System Layout.
- Complies with SVID3 standard.
- Complies with SVR4 specification.
- Refer to the `acct (1M)` man page for more information.
- A symbolic link from the old location of accounting commands (`/usr/lib/acct`) to the new location (`/usr/sbin/acct`) is provided.

Summary and review

- **SAM has improved user interface and many additional features to enable system administrator to manipulate changed components of HP-UX 10.0**
- **Many administrative commands have been added or modified to conform to SVID3 standards or SVR4 specifications.**
- **Additionally, many have been modified to better support Multiple File Systems (MFS)**
- **With HP-UX 10.0, non-root users have access to SAM and to some other administrative tasks that were previously only available to the super-user**

Notes

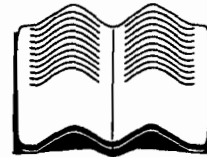
- **For HP employees interested in creating their own SAM-style user interfaces for applications, refer to this module's "Additional Notes" section for a brief description of ObAM, the Object-Action Manager.**

For more information



For more information

- SAM's on-line help
- HP-UX 10.0 release notes
- *System Administration Tasks*
- The man pages for each command



Notes

In this lab, you use SAM to check out new features, and you run new and modified administrative commands.

1. Run SAM in its Motif-based mode.
2. Select *Actions...What's New In SAM*.

What's new with LAN configuration?

Which "What's New..." screen did you use?

What will SAM tell you about an unmounted file system that is about to be overwritten?

Which "What's New..." screen did you use?

-
3. Create a new user.
Specify that the new user must enter a new password at next login.
Login as that user and try it out.
 4. From *Peripheral Devices - Tape Drives*, select displayed tape device.
View *Actions...Create Device Files...Create Custom Device Files* window.
List commonly defaulted parameters:
 5. Go to *Kernel Configuration - Configurable Parameters*.
The "eqmemsize" parameter is new.
What is it?
-

-
6. View *Actions...Apply Tuned Parameter Set*.
Which tuned parameter set(s) are available?

Cancel from dialog box.

7. Re-set the system time to an hour later than it is now,
then set it back.
Which functional area do you use for this?

8. Which performance monitors are available?

9. Go to *Disks and File Systems...File Systems*.
Select *Actions...Add Remote File System...Using the
NFS Automounter*.
Click *Advanced Options* button.
List three parameters you can set from here.

Click *Change Cache Options*.
List the three types of attribute caching.

Cancel from all screens, back to *Disks and File
Systems* area.

-
10. Use SAM to create a logical volume of 80 MB.
From outside of SAM, write some files to your new file system.
Use SAM to delete your logical volume.

11. Go to *Routine Tasks...Find and Remove Unused Filesets*.
Which utility is invoked?

Review its screens, but enter "n" for each option.
Return to *Routine Tasks*.

12. Select *System Log Files*.
List 2 non-ASCII files for which information displays.
-

-
13. Run Restricted SAM Builder.
Assign your new user (created in step 3) permissions to access these functions, in addition to the defaults:
Accounts for Users and Groups
Process Management...Performance Monitors

Login as your new user, and test the restricted SAM.

Leave the restricted SAM running for the next step.

14. As superuser, from different window, invoke SAM.

Go to *Process Management-Process Control*.
Select *Options...Change Refresh Interval*.
Have display refresh every 5 seconds.

Locate your new user's SAM and shell processes.
Leave this SAM window up.

-
15. Have your new user exit restricted SAM.

What happens to the process display?

Have your new user exit the shell.
What happens to the process display?

-
16. From the top level, add this custom application group to SAM: *List Files*

Go to the new group's screen, and add three custom "applications" to it:

Current Directory

Root Directory

Temporary Directory

(Use `/usr/bin/ls`, with appropriate options.)

17. Use SAM's logfile viewer with the following filters:
View yesterday's log only

View today's log only

Summary, then Commands Level

Output to `/tmp/mysamlog` and view it there.

18. Use the man pages, as necessary, to do the following:

Use `groupadd` to add a group: `animals`

What do you enter to do this?

For `useradd`, what does the `-k` option do?

What is the default skeleton directory?

Use `useradd` to add this user to the animals group:

lion

Add a comment: "Roars a lot"

Have it build the home directory for this user.

What do you enter to do all this?

Did the home directory get created? _____

What is it called? _____

19. Run `listusers` to look at the new user.
Run `logins` to look at the new user.
What is the difference between `listusers` and
`logins` output?
20. Use `userdel` to delete lion, and remove its home
directory.
What do you enter to do this?
21. Run `setup`.
What does it allow you to do?
- What is status of password protection?
- Use on-line help to answer the following.
What does "password protection" indicate?
22. Run `sysadm`. What utility is it "really"? _____

-
23. Run `checkfsys`.
What does it tell you? _____
24. Run `fstyp filesystem`, with *filesystem* being
the full name of one of your system's filesystems.
(Run `bdf` to get the name of filesystems).
What is the output? _____
25. Try the following new options to `df`:
-g What information does this tell you? _____
-i What does this tell you? _____
-k What does this tell you? _____
-v What does this tell you? _____
26. Try the following new option to `mount`:
-v What does this tell you? _____
27. What option do you use for `passwd` to force user to
enter new password at next login? _____

This page intentionally left blank.

Additional notes for "SAM and Administrative Commands" module

Object-Action Manager (ObAM)

What is ObAM?

- HP-developed user interface builder/library to prototype and build user interfaces quickly
 - Used to create SAM, SD-UX TUIs and GUIs
 - Customized for developing system administration applications
 - For HP-internal use only

Features

- Provides both X11/Motif and ASCII terminal user interfaces for applications
- Provides automated testing of application and an established help system

Notes

- ObAM development does not require Motif development knowledge.
 - Achieves productivity by sacrificing flexibility. If need full capability of Motif, don't use ObAM.
- Additionally, it provides rapid prototyping of the user interface without C.
 - Callbacks, to enable objects to perform actions, can be added incrementally.
- Requires ObAM Developer's Kit, which can be installed on 10.0 system:
`swinstall -s tomato.fc.hp.com:/release/obam10.0 \*`
- Develop application by writing an ObAM description file.
 - Refer to "ObAM Description File Guide" v.3.0.9 or later for detailed explanation of writing.
 - This guide comes with ObAM Developer's Kit.
- Freely available to internal HP users
 - Not for customers
- Use Mosaic for more information, documentation, examples, FAQs, help:
 - URL is <http://hpfcrcp.fc.hp.com:/obam.html>

The Virtual Memory System



Module goals:

This module shows in detail how the virtual memory (VM) system changed from HP-UX 9.0x to 10.0.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Explain why `vfork()` is faster than `fork()`.
2. Differentiate between swap space and pseudo swap.
3. Tell why process deactivation is better than swapping and describe how HP-UX deactivates processes.
4. Explain how the pager and swapper have been tuned.
5. Describe memory mapped files and the system calls that implement them.

Features and benefits

- **Converge features on S700/S800 platforms**
 - **Memory-mapped files**
 - **Improved process creation**
 - **Expanded limits on process data segment**
 - **Pseudo swap**
- **Improved system performance**
 - **Eliminated process swapping**
 - **New paging algorithms**
 - **Increased process control**

Notes

- **VM subsystem features are now consistent on both platforms**
 - **Memory-mapped files are new for S800.**
 - **A “real” `vfork()` is new for S800, which can be a big performance improver for large processes.**
 - **A special executable type is now also available on S800 to allow programs with very large (> 1 Gbyte) data segments.**
 - **Pseudo swap is now standard on both platforms.**
- **Numerous changes have been made in the paging/swapping mechanisms to improve performance, particularly under heavy system loads.**
- **New process control features have also been added.**

Overview – Converged features

Features adopted from S700

- Memory mapped files
 - Improves I/O performance
 - Provides (yet another) form of IPC
- True `vfork(2)`
 - Child *really* shares parent's VM space
- EXEC_MAGIC
 - Allows for >1GB data segments

Features adopted from S800

- Pseudo-swap
 - Increases VM available to programs

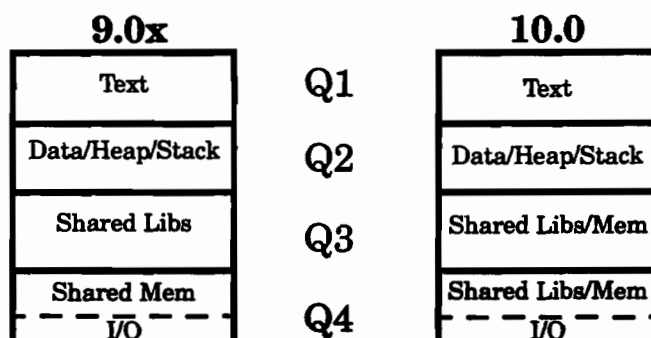


Notes

- Each of these topics is covered in more detail in the following slides.

Virtual address space changes

- MMFs place higher demand on shared memory
- Shared memory use more flexible
 - Shared segment is now 1Gb + 0.75 Gb (Q3+Q4)
 - Shared libs, shmem, and MMFs all stored together



Notes

- Previously, shared libs and MMFs were stored in Quadrant 3, and shared memory was restricted to Quadrant 4
- Now, Quadrants 3 & 4 have been combined, and all types of shared data (libs, mem, and MMFs) are stored there. Note that the last 250 Mb of Q4 are still reserved for I/O addressing, and that there is a gateway page between Q3&4, so the maximum size of a single segment is 1Gb.

Memory mapped files

- Enhanced functionality from 9.0x S700
 - Complies with standards: POSIX, OSF AES
 - `madvise()` “works”, read-ahead implemented
 - Memory mapped device files not supported
- Maps file content directly into program space
 - Bypasses buffer cache mechanism
 - Program controls sync-ing of contents
- Allows faster access to files
 - Smaller (<8k) reads/writes 10-100x faster
 - Larger (~8k) reads/writes marginally faster

Notes

- MMFs work by allowing a program to map file system files directly into their program space, thus bypassing the buffer cache. This can improve performance, especially for small (<< file system block size) reads and writes. MMFs are faster because:
 - They avoid the redundant data copy to and from the program's space to the buffer cache when reading and writing the file.
 - For randomly dispersed I/O (in contrast with sequential), you avoid the overhead of `lseek()`.
- MMFs are of limited usefulness because:
 - You cannot grow a memory mapped file unless space is acquired at the time of the mapping.
 - A program must manage file synchronization itself.
 - Files shouldn't be accessed via the file system and MMFs simultaneously.

Programming calls for MMFs

- Include the system header file:

```
#include <sys/mman.h>
```

- Calls for creation, manipulating MMFs

```
mmap(addr, len, prot, flags, fildes, off);  
madvise(addr, len, behavior);  
mprotect(addr, len, prot);  
msync(addr, len, flags);  
munmap(addr, len);
```

- Calls for MMF semaphores:

```
msem_init(*sem, initState);  
msem_lock(*sem, blockOrReturn);  
msem_unlock(*sem, merelyCheck);  
msem_remove(*sem);
```

Notes

- For `mmap()`, the *flags* can be: `MAP_FILE`, `MAP_ANONYMOUS`, `MAP_VARIABLE`, `MAP_FIXED`, `MAP_SHARED`, `MAP_PRIVATE`.
- For `mmap()` and `mprotect()`, the protection flags can be `PROT_READ`, `PROT_WRITE`, `PROT_EXEC`; or `PROT_NONE` will deny all access.
- The *behavior* flags for `madvise()` influence the actions of MMF internals for random I/O (to read smaller amounts of data) versus sequential access (to read more data and read ahead).
- For `msync()`, the *flags* can be `MS_SYNC` or `MS_ASYNC` for style of writing, and `MS_INVALIDATE` to mark the memory as out-of-date.

The `vfork(2)` system call

- Efficient way for applications to spawn a small subprocess

<code>fork()</code>	vs.	<code>vfork()</code>
Kernel allocates swap for child with same size as parent		No extra swap allocated; <i>child shares swap with parent</i>
Kernel copies parent's text space only to immediately overlay it		Child shares text with parent; no copying
Parent and child have separate data spaces		Parent and child <i>share data spaces</i>
Parent timeslices with child		Parent sleeps while child runs

Notes

- In previous versions of HP-UX, `vfork()` on the S800 didn't implement true `vfork()` semantics, but it was actually implemented as `fork()` w/ copy-on-write for the data segment of the child. On the S700, `vfork()` has not changed from the 9.0x release.
- `vfork()` is faster than `fork()`:
 - The kernel does not create a virtual address space for the child; the child shares text and data with the parent.
 - The kernel does not reserve extra swap space for the child process.
- Because the parent and child share data, including I/O descriptors, the child can write into global variables and the parent will see the change. This is potentially *very* dangerous! So, the child of a `vfork` must follow some stringent rules:
 - The child must immediately call `exec()` after `vfork()`.
 - If the child cannot `exec()`, it should call `_exit()`, *not* `exit()`!

EXEC_MAGIC executable

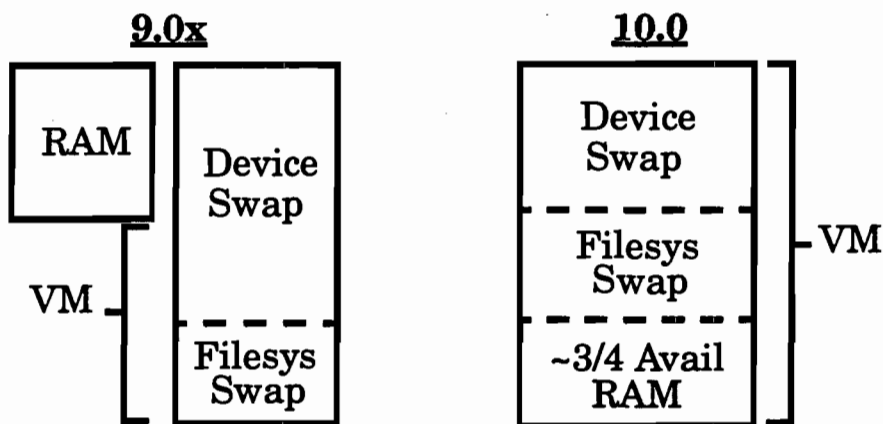
- In 9.0x program data segment size was limited to 1 Gb (really 945 Mb)
 - Some large applications need more than this
- Now, EXEC_MAGIC programs can have ~2 GB of combined *text* + *data* + *bss*
 - Loader option (-N) creates the EXEC_MAGIC executable
- Issues/restrictions/limitations
 - Program text is not shareable
 - Entire text+data segment is swapable
 - Text+data faulted in at process start-up

Notes

- This functionality was first introduced at HP-UX 9.0 on the S700 platform. It is now offered on both, and is the same.
- Text and data (Q1 & Q2 of the process virtual space) are combined into one big data object in an EXEC_MAGIC program. The data segment immediately follows the text segment, rather than starting at the beginning of Q2, as it normally would.
- The maximum size of the text and data segments of an EXEC_MAGIC executable is 1,969 Mbytes.
- Because of the unusual format of the virtual address space for this type of executable, there are some restrictions on them:
 - EXEC_MAGIC programs cannot be marked as shared text.
 - At `exec()` time, the entire image is faulted into memory instead of a page at a time
 - The kernel reserves swap for the text *and* data spaces instead of just the data — more potential impact on swap

Changes to VM Model

- Requirement that all writable pages of process be mapped to backing store removed at 10.0



Notes

- In HP-UX 9.0x, a system with 1 Gb of memory but only 500 Mb of swap could create only 500 Mb of processes.
 - HP-UX guarantees swap space for every process, even though the process might never need to be swapped out
- At swapon time, the newly-available disk space is converted to units of pages and added to the data structures in the kernel. For dynamic swap, the minimum specified is added.
- Processes reserve swap at creation and when growing, as in when making a call to `malloc()` or `sbrk()`. Swap usage shrinks at process termination and process shrinkage. When the kernel is out of swap space on disk, no more processes can run, and none of the running ones can grow.

Pseudo swap

- Memory that masquerades as swap space
 - Doesn't change pre-10.0 functionality; processes that need to be swapped will be
 - Up to 3/4 of available RAM will be used
- Total VM available to processes is now:
 $\text{dev. swap} + \text{file sys. swap} + \sim 3/4 \text{ of avail. RAM}$

Note: Ordering of above is purposeful; pseudo swap is used only as a last resort

- Since more processes may now exist simultaneously, this may affect performance

Notes

- Swap space is *reserved* at process creation time, but not *allocated* in the swap space until pages need to be pushed out to disk. For a system with 64 Mb free, the kernel will track 48 Mb of memory as being a candidate for being used as pseudo swap.
- Once all of physical swap and pseudo swap is exhausted, attempts at process growth or creation will fail.
- Can be disabled by setting kernel tunable, `swappmem_on`, to zero (0).
- The `statd` daemon migrates pages reserved on pseudo swap to the physical swap (disk) as space there becomes free. This ensures that "real" swap space is used if it can be.

Paging system changes

- Changes to vhand
 - Bias paging to favor interactive processes
 - Treat multiply-referenced shared pages fairly
 - Treat small and large preions equally
 - Consider system I/O load when paging
 - Adapt paging to system characteristics
- Changes to swapper
 - Eliminate swapping of processes
 - Implement “proper” deactivation
 - Allow for serialization of processes under heavy system load

Notes

- The pager, named vhand, runs when memory gets low (less than **gpgslim** pages). It scans the pages of processes (by preion), marking them as unreferenced. Pages that not referenced between the time they are marked and the time they are to be “stolen” are paged out to secondary storage.
- In 9.0x, the paging and swapping routines had a number of characteristics that caused system performance to suffer under high memory-pressure conditions. For instance:
 - The vhand daemon looked at pages at the same rate, regardless of the size of the preion. Thus, a page in a 1 Mb preion was 100 times more likely to be paged out than a page in a 100 Mb preion.
 - Shared pages, for example shared text regions, were put on the list once for each process that reference them. A shared memory area (data preion) shared by 30 processes was thus 30 times more likely to be paged out than a non-shared preion of the same size.
- Many changes have been made to the paging and swapping algorithms to address these and other problems.

Changes to vhand

- Process smart
 - Age pages for “nice” processes more frequently than those of “normal” processes
- Fairer
 - Put shared pages on active list only once
 - Scan fixed portion (1/16) of preregion at a time
 - Start scan in preregion where last ended
- I/O load smart
 - Reduce paging rate if disk queues fill
 - Page only enough pages to reach lotsfree
- CPU smart
 - Increase hand spread on fast/MP systems

Notes

- The algorithms in vhand now consider process nice value in their determination of which pages to steal. This is done to favor interactive processes over batch jobs.
- A page is only on the active list once, regardless of the number of processes that reference it.
- Paging begins as it did in the past, when the amount of free memory drops below the **gpgslim** threshold. Preregion scans begin at the point in the region where they last ended within a preregion in HP-UX 10.0. Also, a fixed percentage of the region is scanned, regardless of its size.
- The pager keeps track of how much memory will be freed up after scheduled I/Os complete, and uses the **lotsfree** value as a threshold, so it knows when to stop stealing. It also monitors the disk queues to avoid overloading the I/O subsystem.
- It's also able to modify things like the age/steal hand spread in order to take system-specific characteristics into account. It even accounts correctly for MP systems.

Changes to the swapper (sched)

- Processes are no longer swapped
 - When memory pressure is high, swapper deactivates low priority processes
 - Paging routines do the rest
- Deactivated process is removed from run queue, its pages are immediately aged
- Factors in deciding who to deactivate/reactivate
 - Process priority
 - Process size
 - Process state
 - Process type

Notes

- Process deactivation/reactivation **replaces** process swapping.
- The swapper (sched) monitors free pages, and when the pool drops below the **minfree** threshold, it looks for processes to deactivate. When a candidate is found, it is removed from the run queue, and its pages are immediately aged so that they'll be immediately stolen.
 - Processes that can't immediately be deactivated (because they're doing an I/O or hold a lock, for instance) are set-up so that they will be self-deactivated at the next convenient moment.
- Process that are nice'd are more likely to be deactivated, so nice'ing (or renice'ing) batch jobs may have a positive affect on interactive performance.
- The pager watches how frequently transient conditions drive free memory toward zero, and adjusts its threshold to be an appropriate figure for the current workload. This ensures the definition of "high memory pressure" is appropriate, and also prevents the system from overreacting to transient conditions.

serialize and renice

- Serialized programs run one after another, but only when memory pressure is “high”
 - Will run as regular processes at other times
 - User/admin has no control over order of serialization once it takes affect
 - Serialization will stop if system load drops
- To serialize a process, use `serialize`

```
$ serialize commandName commandArgs
$ serialize -p pid
$ serialize -t -p pid
```
- Also there's a new command, `renice`, to change the nice value of a process that's already running

Notes

- Serialization is another swapper (sched) trick. When memory pressure becomes “high” – less than desfree free pages, or high paging activity with little CPU utilization – all processes that have been marked for serialization get put into a special queue. From that time forward, they act as if only one of them were running, effectively reducing system load. When the first job on the serialized job queue finishes, the next will start. This continues until all the serialized jobs complete, or until system memory pressure drops, at which point the serialized jobs act like regular processes again.
- A few notes about serialization:
 - There is no way to tell later if a process is serialized.
 - The user/administrator has no special control over when serialization starts, nor the order in which jobs serialize.
- In order to provide more control over processes, there is a command that allows the nice value of already-running processes to be changed. It's called `renice`.

Summary and review

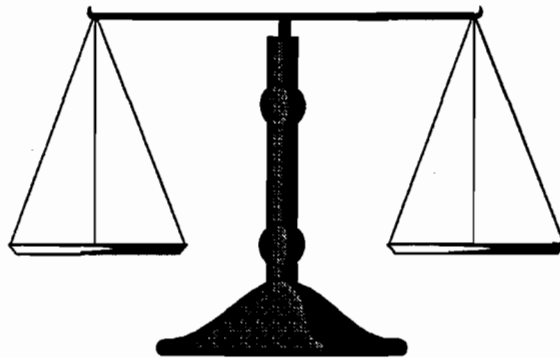
- Converge features on S700/S800 platforms
 - Memory-mapped files (MMFs)
 - Improved process creation (vfork())
 - Expanded limits on process data segment (EXEC_MAGIC)
 - Pseudo swap
- Improved system performance
 - Eliminated process swapping
 - New paging algorithms (vhand/sched)
 - Increased process control (serialize/renice)

Notes



This page intentionally left blank.

Process Resource Manager (PRM)



Module goals:

The purpose of this module is to introduce the Process Resource Manager (PRM), its basic operation, administration, and utilities.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe PRM and why it is useful.
2. List the PRM commands.
3. Configure and activate PRM.
4. Change PRM dynamically.

What is PRM?

- Process scheduler, commands
- Manages CPU allocation for groups of users
 - Guarantees a minimum percentage of the CPU
 - You can dynamically change CPU allocation
- Optional product: Customers pay extra for it
 - Is it loaded? Run `prmconfig` (or `ps` or `prmmmonitor`)
- Not real-time, but for the commercial market
 - Mainframe alternative

Notes

- PRM is meant to maintain, say, a ticker-tape reader despite a system load that swings wildly, where ordinarily some data could be lost.
- PRM is *not* a real-time (RT) scheduler. Intended for commercial markets, it allows HP-UX to be sold as an alternative to a mainframe.
 - It can control RT processes, but violates RT scheduling ideas like interrupt latency
 - A RT program is permitted to exceed its fair share, but other processes in the same resource group will be starved until the RT program gives up the CPU

Why use PRM?

- Control use of CPU as a resource
 - Create a process resource group, and put a job into that group
 - Guarantee that a group gets at least a certain percentage of the CPU
- Budget CPU resource
- Monitor CPU usage
- Complete critical tasks on time regardless of the system's load

Notes

- You can dedicate 50% of the CPU to processing the payroll even though the backup normally consumes 75% of the system — the payroll will get its 50% and the backup will just take longer.
- A clarification: Resource groups are not related to the group entries in `/etc/group` or `/etc/passwd`. They are defined in separate files and are separate. While the administrator may choose to create a correspondence between the group-ID field from `/etc/passwd` or `/etc/group` and the PRM configuration file, the fields are not necessarily related.

Standard HP-UX scheduler

- **Priority-based round-robin scheduling**
 - Round-robin scheduling at each priority level
 - Each process has a priority that changes
 - Process with highest priority gets launched
- **Change the scheduling algorithm with:**
 - Real time (priority stays constant)
 - Starting a process with a nice value (priority starts lower, stays lower)

Notes

- **Round-robin means:**
 - There is a logical queue at each priority level
 - A process runs until:
 - It gives up the CPU (sleeping on a resource, for instance), or
 - It reaches its time limit, or
 - A process with a higher priority is ready to run.

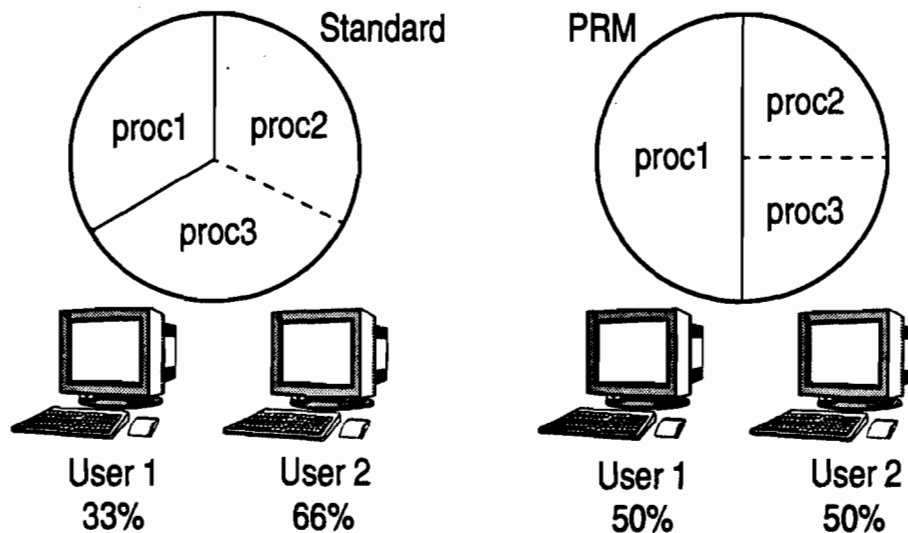
The PRM scheduler

- Runs on top of the normal scheduler
- Each user has access to one or more PRM groups
 - Defined and controlled by system administrator
 - Each group is guaranteed a minimum percentage of the CPU
- A process is assigned to a group
 - Defaults to an initial group
 - A process is in only one group
- Within a group, standard CPU scheduling policies apply (round-robin, ...)

Notes

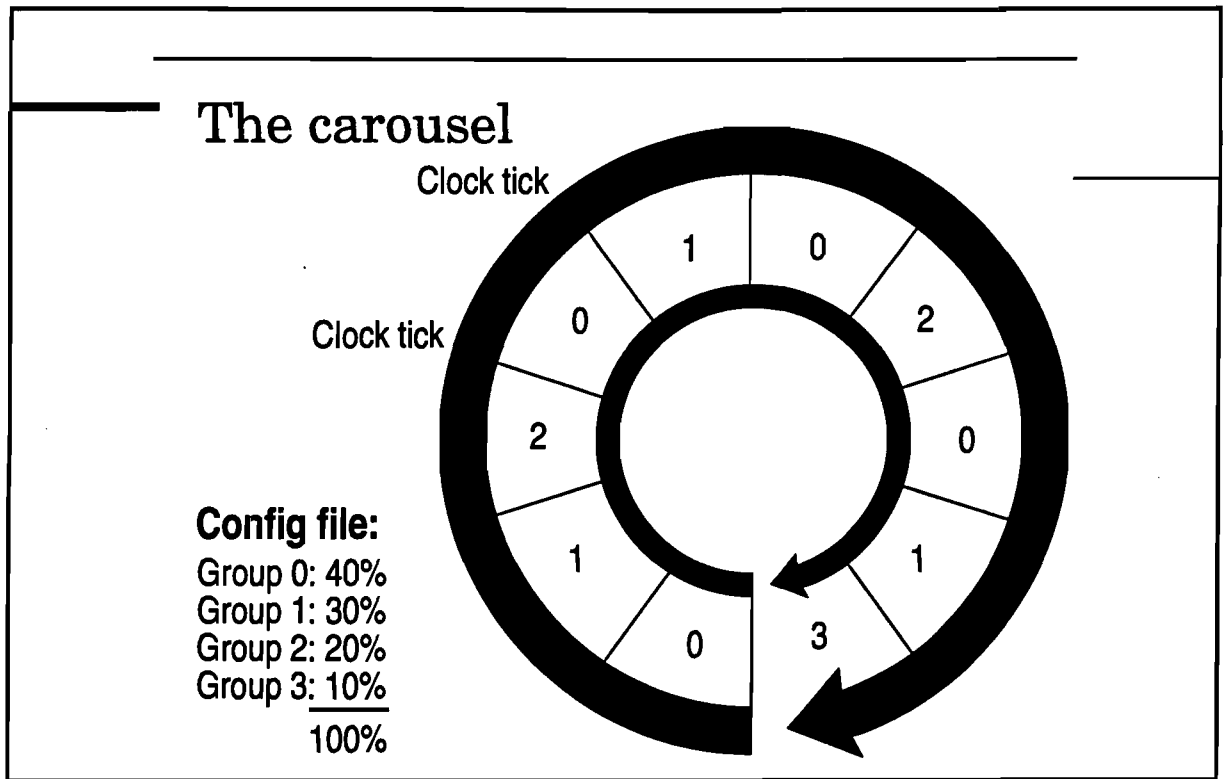
- Actually, PRM replaces the scheduler by patching the system call dispatch table. Within a PRM group, PRM behaves like the standard scheduler.

How PRM works



Notes

- **Standard scheduler:** He who runs the most programs wins (gets the CPU).
- **PRM:** Divide the CPU among the groups. Each group gets its fair portion.
 - If a group is not using the system, the other groups get the extra CPU according to their relative proportions
 - If the CPU has idle time, there are enough cycles to go around
 - Because there is no contention, the percentages may not look like the defined proportions



Notes

- PRM divides the carousel into 100 horses, or slots. It then goes around the carousel to load up the horses, or in other words assign each slot to a group according to the relative percentages.
- A clock tick happens every 10 milliseconds as a hardware interrupt. When it ticks, PRM moves its pointer to the next horse.
- The scheduler runs the highest-priority process on the current horse (meaning the current resource group). If none in that group are ready to run, the scheduler advances to the next horse.
- PRM handles multi-processor systems:
 - It still uses a 100% model; the percentages must still add up to 100
 - There is a carousel for each processor
 - Within a process group, PRM tries to distribute software processes across hardware processors

Installing PRM

- **Install using SD-UX**
 - **Requires kernel rebuild and reboot**
 - **Configuration file is automatically built from /etc/passwd file (logins go into default user group)**
- **After install:**
 - **PRM is in “reset” state**
 - **PRM disabled (operating system is using standard HP-UX scheduler)**

Notes

- **Because the configuration file is based on the /etc/passwd file, how does this affect subsystems that bypass this file, such as:**
 - **NIS?**
- **Users not listed in the PRM configuration file will be placed into group 1.**

Configuration strategy

- Two models of PRM configuration are possible:
 - Resource allocation budget
 - Management by exception
- 1. Define resource groups
 - How many groups are needed?
16 max (system & user defaults take 2 of them)
 - What percentage does each group get?
- 2. Set users into groups
 - Initial resource group
 - Additional groups in which they can run jobs

Notes

- To use a resource allocation budget, figure out the workload mix and divide the users into groups ahead of time. For example:
 - Payroll, bookkeeping, forecasting
 - Stock market data gathering, stock market analysis
 - Accounting, marketing, human resources
- To use management by exception, leave everyone in the default user group and pull them out into separate groups over time.
- The administrator can define more than one configuration (for example, month-end and month-middle), then switch it with cron:

```
cp /etc/prmconf.end /etc/prmconf && prmconfig -k
```

Configuration tactics

- 1. Become super-user**
- 2. Edit /etc/prmconf with a text editor**
 - Establish groups and percentages**
 - Establish user access restrictions**
- 3. Run `prmconfig -sei` to perform supplemental checks, enable the scheduler, and move processes into their initial groups**
- 4. Edit /etc/rc.config.d/prm so the PRM start-up script will run when the system reboots**

Notes

- **Because `prmloadconf` reads the `/etc/passwd` file (only):**
 - Administrators of NIS systems should manually add the real users from the password map.**

The configuration file
`/etc/prmconf`



The configuration file

One `/etc/prmconf` file, two sections

- PRM group records
`groupName:prmID:cpuPercentage::`
- PRM user records
`userLogin::::initialPrmGroup[,otherPrmGroup,...]`
- Note four colons in each record
- Comments
`# This is a comment`

Notes

- The configuration file defines groups and their percentages.
- The administrator can define up to 16 resource groups, although the first two (0 and 1) are defined by PRM.
- PRM group records consist of:
 - Number or PRMID
 - Name
 - CPU entitlement (minimum percentage)
- Two group numbers are reserved:
 - ID 0, the PRM system group
 - ID 1, the user default group
- The user records contain an initial group followed by optional alternate groups.
- HP recommends that group 0 (PRM system group) should have 20% or more. PRM enforces 20% as a minimum.

The simplest configuration file

- Effectively just two lines:

```
PRM_SYS:0:20::
```

```
OTHERS:1:80::
```

- Super-user's initial group is default user group, thus there is an implied third line:

```
root::::OTHERS
```
- Other logins are added from `/etc/passwd` when the product is installed
- For root and super-user records, only the initial group will be used

Notes

- A sample customized configuration file:

```
##### Section 1 -- Groups #####
# System processes, boot processes, super-users
system:0:20::
# The default users will share the CPU with the data-crunchers
default_users:1:25::
# Group for processes that crunch all that data
crunchers:2:45::
# (Nearly-) Spare cycles for those doing data modeling
forecasters:3:10::
#
##### Section 2 -- Users #####
root::::system
donsu::::crunchers
don::::crunchers
cayce::::forecasters,default_users
```

Configuration file utilities

- `prmloadconf`
 - Loads user records from `/etc/passwd`
 - Can run it more than once to add new logins
- `prmconfig -s`
 - Performs supplemental checks
 - Compares users listed in `/etc/passwd` and `/etc/prmconf`
 - Reports mismatches, but allows you to configure PRM with the file anyway

Notes

- You can expect `prmconfig -s` to note mismatches if you don't specify all users in the configuration file.

PRM states

State	Established by	Result
Reset	System reboot or <code>prmconfig -r</code>	No resource budget, standard HP-UX scheduler
Configured	<code>prmconfig -d</code>	Resource budget loaded, standard HP-UX scheduler
Enabled	<code>prmconfig -e</code>	Resource budget loaded, PRM scheduler is active

Notes

- The configured state can also be established:
 - From the reset state with `prmconfig -i` or `-k`
 - From the configured state with `prmconfig -i` or `-k`
 - From the enabled state with `prmconfig -d`
- The enabled state can also be established:
 - From the reset state with `prmconfig -ie` or `-ke`

The prmconfig command

- Configure, enable, and disable the scheduler
- Check state and configuration information

Option	Meaning
-i	Configure or reconfigure PRM from the config file & initialize
-k	Configure or reconfigure PRM from the config file & keep
-s	Checks the config file for problems, including syntax
-e	Enable the PRM scheduler
-d	Disable the PRM scheduler
-r	Reset: Disable the PRM scheduler & ignore the budget

Notes

- The start-up scripts use `prmconfig -ie` instead of `prmconfig -ke`. The `-k` option keeps running processes in their existing resource-group assignments, but `-i` moves the processes to their user's initial resource group. The manual page mentions the exceptions to this general rule.

Dynamic changes

- Change PRM group of processes with `prmmove`
- Change the active configuration
 - Add or delete resource groups
 - Change PRM groups (e.g., percentages)
 - Change users (initial group, alternate groups)
 - Steps:
 - # `vi /etc/prmconf`
 - # `prmconfig -k (or -i)`
- With `-k`, processes continue to execute in current resource group until user logs out or:
 - # `prmmove newGroup -u userLogin`

Notes

- PRM uses a *copy* of the configuration file created at `prmconfig` time, so you can accidentally mangle `/etc/prmconf` without destroying the running PRM scheduler.

The prmmove command

- Move a process to a different resource group
- Move a user temporarily into another group

prmmove *where what*

- *where*

A resource group name or ID

-i User's initial group

- *what*

A program with its options and arguments

-u All processes belonging to a user

-p An individual process

-g All members of a process group

Notes

- Examples:

```
prmmove forecasters betmetric -eoyr statistics.q[1-4]
prmmove system -u donsu
prmmove crunchers -p 8318
prmmove 0 -g 4310
```
- The command form (first example above) is a one-time case of “run this program in the specified resource group.” It does not affect future runs of the program.
 - For instance, there is no way to say, “Run all SAS processes in the crunchers group, but run all FrameMaker jobs in the pubs group”
 - Of course, someone could write a wrapper script or alias:

```
alias frame='prmmove pubs $FMHOME/bin/maker'
```

but users of this alias/wrapper would need the pubs group defined as an initial or additional group in their user records in the configuration file
- The -u, -p, and -g options can take one or more arguments.

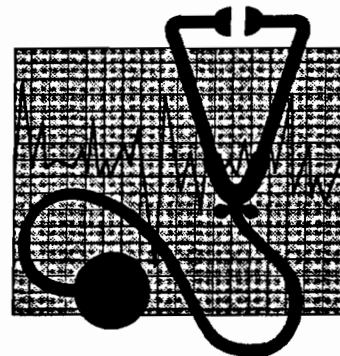
PRM-smart commands

Command	Description
ps, id,	-P option to show PRM ID
acctcom	-R option to show info about a resource group
at, cron	Place jobs in user's initial resource group
login	Place user's shell in user's initial resource group
fork()	Child inherits process resource group of parent
exec()	Process resource group stays the same

Notes

Monitoring CPU usage

- The `prmmmonitor` program
 - Monitor CPU usage at resource group level
- Glance, PerfRX
 - Global resource usage
 - Process resource usage
- Other system tools
 - `ps`, `top`, `sar`, `acctcom`



Notes

The prmmmonitor command

- `prmmmonitor [-t] [interval [iterations]]`
-t Terse output: Don't print headers
- Similar to `iostat`, `vmstat`, `sar`:
If you don't specify number of iterations,
interrupt (^C, etc.) to stop



Notes

- Example: Run every 5 seconds for a minute:
\$ `prmmmonitor 5 12`

Performance impact

- PRM measured to take less than 1% of CPU
- If configuration file is large (> 1 Kb), commands that report or use PRM configuration data will slow down

Notes

- PRM-aware commands are `ps`, `id`, `login`, `prmmmonitor`, and `acctcom`. Most PRM commands and PRM-aware commands read the file.

-
1. On the S700, create a logical volume of 100 Mb to hold the files used to exercise the system. Type in the following shell commands.

```
lvcreate -L 100 -n lvcint vg00
newfs -F hfs /dev/vg00/rlvcint
mkdir /CINT92
mount /dev/vg00/lvcint /CINT92
/opt/deltaclass/bin/unpack-CINT92
```

Note: These benchmarks are licensed. It is a violation of our license to copy them to a non-HP site. Do NOT take this software out of the classroom.

2. Install the optional PRM product. Get the software from the same machine that provided your operating system.
3. Set up accounts for the following users (put them in the default login group of users). Use SAM, which will take care of details such as creating the home directories. In SAM's *Add a User Account* window, press the *Apply* button instead of *OK* to keep the window up until you add all three accounts.

```
frost
ember
torch
```

4. Does the system's PRM configuration file contain the new logins?

Run the command to load the `/etc/passwd` file into the PRM configuration file. Record your command:

-
5. Set up the configuration file to define the following. Copy your PRM configuration-file information to this _____ page.

PRM Group	CPU
system	50%
users	15%
hot	30%
cold	5%

User	PRM Group (instead of the default user group)
frost	cold
ember	cold as initial group but enrolled in hot too
torch	hot

6. Run the `ps` command with a PRM option to find out quickly whether PRM is running or not. (For this and the rest of the exercises, record your commands after the questions or in the margins.)
7. Completely activate PRM, performing the extra supplemental checks when you do so.
8. Run a PRM command to verify that PRM is working.
9. Simulate a busy system administrator: Add another account named `luke` to the system with SAM, but do not add that account to the PRM configuration file. (In

the crush of work, you just forgot.)

10. Start jobs by the different users. Type these commands in each user's account. Instead of logging out and logging in again, launch a window on the S700 for each account with the commands:

```
# PATH=$PATH:/usr/vue/bin
# alias dt="dtterm -fg PapayaWhip -title"
# dt frost -bg MediumSlateBlue -e login frost &
# dt luke -bg Gray45 -e login luke &
# dt ember -bg RosyBrown4 -e login ember &
# dt torch -bg PaleVioletRed4 -e login torch &
```

Then, after logging in as the appropriate user, type the following in each window (start with torch):

```
$ cd /CINT92
$ ./runme
```

11. As the super-user, verify that PRM is giving each user a proper percentage of the CPU.
12. As user ember, interrupt (kill) your current job. Then run the runme program in the PRM group hot. (Recall that ember's primary group is not hot.)
13. As user frost, interrupt your current job. Then try to run runme in the PRM group hot.

Still as frost, try to run runme in the default PRM

user group.

14. As user luke, interrupt your current job. Try running `runme` in the `hot` group. What prevents the command from working?

Still as luke, run `runme` in your default PRM group.

Then, as the super-user, move the currently-running jobs of luke into the PRM group `hot`.

15. User ember notices that she can promote her jobs to the group `hot` with one command. Interrupt her current job and type that command into her window. Then run the `runme` job again, and verify with `ps` that it is running in the `hot` group.

16. Interrupt luke's job (running in the `hot` group) and start another in his normal PRM group.

17. As the super-user, you notice that some of the users on the system (like luke) are getting more of the CPU than the paying customers in the group `cold`. Why?

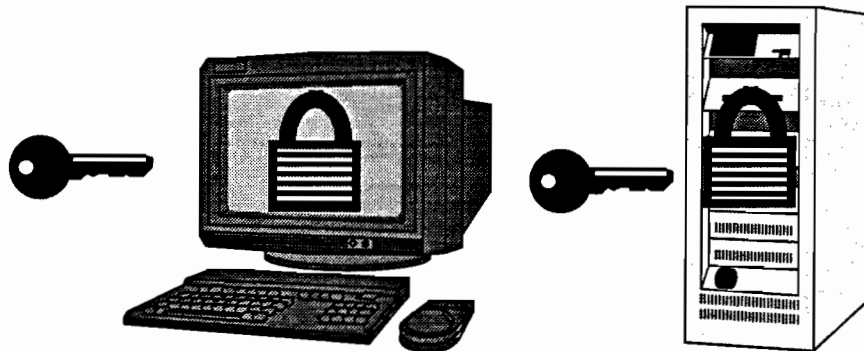
One solution is to specify luke in the correct PRM group. Another solution is to change the PRM budget so that the default user group gets 5% and the `cold` group gets 15%. Do this second solution, writing down the changes you make to the configuration file. Then write down the command you used to reconfigure

PRM with the new settings.

18. **Verify that your changes worked.**

19. **For the commands `id` and `ps`, run them:**
 - A) **Without any PRM options**
 - B) **With the `-P` option**
 - C) **With the `-R` option (if it applies)**

Trusted Systems



Module goals:

The purpose of this module is to describe the new and enhanced features of trusted systems under HP-UX 10.0.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Describe the motivations for changing trusted systems.
2. List and describe new features and benefits of trusted systems.
3. Explain the changes to the protected password files.
4. List the password generation options.
5. Explain the phases and results of password aging.
6. Describe the differences between time-based locking, account-locking and login control and location-based locking.
7. Recognize the new and obsolete developer calls used with trusted systems.

Motivations for changing

- Satisfy users' needs for better and more stringent password management and logon restrictions
 - Provide features similar to the BLS product
 - Continue to provide features of C2 security level auditing
 - Features of C2 password management that were used on 9.0x no longer supported



Hacker's Codebook

Notes

- The new functionality provides added security compared to a 9.0x system.
- The functionality added adheres to Department of Defense's "Green Book" requirements, which is also followed by a BLS.

Features and benefits

Features

- Password generation and screening
- Password aging
- Single user boot authentication
- Time-based access controls
- Account locking and login control
- Location-based access controls

Benefits

- System administrator has greater control over
 - password security and usage
 - who can access what system, at which time, from which location



Notes

- Most of these features are discussed in more detail on the next pages.
- Single-user boot authentication refers to the ability to have a user and password for being able to boot or reboot.

Protected password file changes

- Protected password database files stored in
/tcb/files/auth hierarchy
- User authentication profiles stored in:
 /tcb/files/auth/user_initial/user_name
 – e.g., for user: joebob
 /tcb/files/auth/j/joebob
- Old file obsolete: / .secure/etc/passwd
- System default password data stored in:
 /tcb/files/auth/system/default
- Many new authentication entries in protection
password database files, based on new features
 – See prpwd(4) man page for more information

Notes

- Protected password filenames based on login names.
- The password database files have many new entries reflecting the new features available for assigning to users on a trusted system.
- For example:
 - Password lifetime
 - Whether password is generated by system or user
 - Terminal or remote host associated with most recent successful login
 - Terminal or remote host associated with most recent unsuccessful login
- See the new man page prpwd(4) for a list of all of the entries in a protected password file.
- These files should never be directly edited. Use SAM to make changes. The files will not be maintained otherwise.

Password generation

- System administrator specifies maximum length
- User-generated
 - Screening options available
- System-generated
 - Letters only
 - Letters, numbers and punctuation characters
 - Pronounceable phrases (English only)
- Password generation options per-system, per-user
- Each user must have at least one password generation option set
 - If more than one option, password generation menu displays when changing password

Notes

- System administrator specifies maximum password length, up to 40 characters, for each user account.
- Password generation can be specified on a system-wide basis.
 - Use SAM to do this.
- Additionally, the system administrator can override on a per-user basis, or can just set it per-user.

Password screening

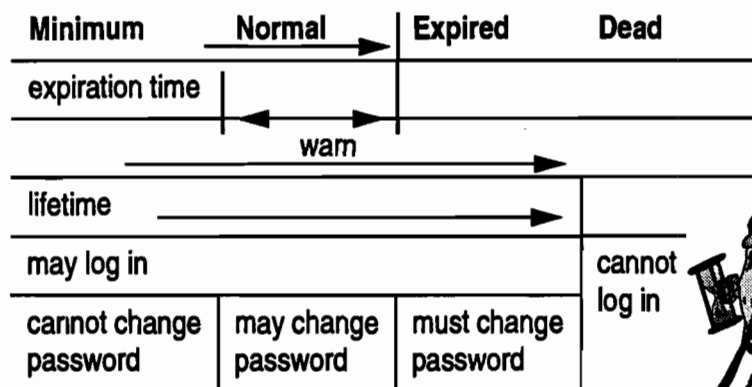
- The optional restricted password feature can be used to screen new passwords.
- It will not allow:
 - Dictionary matches
 - Login names (unless contains number)
 - Permutation of login names
 - Repeated characters
 - Palindromes
 - Nulls



Notes

Password aging

- Aging periods set for number of days, not weeks
- Can set system-wide or user aging periods
- Phases and events at each phase:



Notes

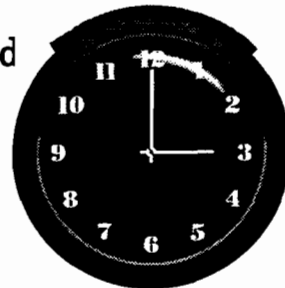
- When password aging is set, each password goes through a series of phases from the time it is first defined.
- Password aging is available for non-secure systems also, but there are no warnings given to users on non-secure systems.

Phase	Notes
Minimum	Forces user to use new password. Intent is so that user will not change back to old password.
Normal	If user changes password during this phase, cycle starts back at minimum. At end, warning message on impending password expiration displays after login
Expired	User must change password during logon, or login is terminated.
Dead	If user does not change password before this, account is locked to user. System administrator can remove lock and reset password.

- Use SAM to control for system-wide aging.

Time-based access controls

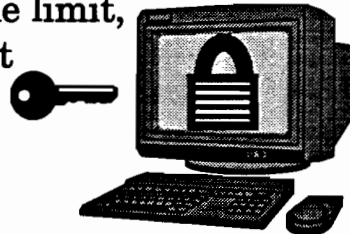
- Set day-of-week and time-of-day that a given user can log on to system (per-user feature)
 - Monday through Friday
 - Entire week
 - Specific days of week
 - Time can be any interval during 24-hour day
- If user attempts login outside of allowed period, login is terminated
 - Event is logged, if auditing on
- Superuser can login outside allowed period
 - Event is logged



Notes

Account locking and login control

- Disable login on repeated unsuccessful login attempts
 - Number of unsuccessful login attempts allowed can be specified
- Enforce time delay between login attempts
 - Delay between login attempts can be specified
- After specified number of days of inactivity, or when reaches an absolute time limit, (e.g., 12/31/95) disable account
- Use SAM to control



Notes

Location-based access controls

- Specify which users allowed on terminal or from host
 - List users in Device Assignment Database, `/tcb/files/devassign`
- Use terminal database file `/tcb/files/ttys` for:
 - User and timestamp of most recent successful and unsuccessful login attempts for terminal
 - Delay imposed between login attempts from terminal
 - Number of unsuccessful attempts allowed before locking terminal



Notes

- Terminals are the only device that can be restricted in this manner.
- The information is specified for each terminal that can be used to login to the host.

New and obsolete developer calls

Call	Status	Purpose/Replacement
getprdfent putprdfent	New	Manipulates system defaults in /tcb/files/auth/system/default
getprpwent putprpwent	New	Manipulates password entries from /tcb/files/auth/*/*
getprtcent putprtcent	New	Manipulates terminal control database /tcb/files/ttys
getprdvagent putprdvagent	New	Manipulate device assignment database entry in /tcb/files/auth/devassign
iscomsec	New	Verifies whether or not commercial security is installed
putspwent	Obsolete	Replaced by putprpwent
fgetspwent	Obsolete	Replaced by putprpwent

Notes

- Refer to the man pages for syntax and usage.
- Obsolete calls are not supported on HP-UX 10.0 trusted systems.
 - They remain on the system for usage by those customers who have created their own shadow password file.
 - The functionality of both can be replaced using the putprpwent call.

Tips and tricks

Problem	Your reply involves
We had a secure system. I installed DCE. Now our secure system is so secure, it won't let anyone login.	DCE replaced <code>login(1)</code> , but <code>login(1)</code> and <code>passwd(1)</code> are specific to security. Restore the two commands from a pre-DCE backup.
No NIS (yellow pages) support	(Same as 9.0x)

Notes

Summary and review

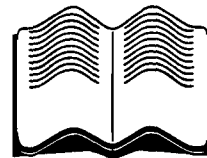
- Trusted systems allow new features of:
 - Password management
 - generation by the system
 - screening of user-generated passwords
 - control of the lifetime of passwords
 - Control of account logins
 - when a user can login
 - delays between repeated logins
 - length of inactivity between logins
 - number of unsuccessful logins allowed
 - Control of terminal logins
 - which users can access
 - delays between access
 - number of unsuccessful access attempts

Notes

- Additionally, there are new man pages, and new development calls.

For more information

- New man pages:
 - devassign(4)
 - default(4)
 - authcap(4)
 - prpwd(4)
 - ttys(4)
- The “System Security” chapter in *System Administration Tasks*



Notes

- Additionally, refer to the changed passwd(1) and passwd(4) pages.

1. What is the pathname to protected password files?

2. If the optional screening password feature is on,
which of these would not be acceptable? Why not?

Attempted password	Acceptable?	If not, why not?
abcba		
abcbaf		
root5		
root		
madam		

3. If the password is in the “normal” phase, and the user
changes the password, at what phase is the
password?

4. What are the three major ways to control user logins?

This page intentionally left blank.

Course Review

Module goals:

The purpose of this module is to summarize and review the entire course, and to provide the student with information about other resources to continue learning about HP-UX 10.0.

Student performance objectives:

At the conclusion of this module, the student will be able to:

1. Summarize the entire course.
2. List additional resources to learn more about HP-UX 10.0.

Summary and review

- **Pre-upgrade analysis**
- **Installation of HP-UX**
- **File system layout**
- **User environment**
- **System startup**
- **I/O convergence**
- **SD-UX software management**
- **Packaging with SD-UX**
- **LVM**

Notes

- **For class discussion:**

Next week, a colleague asks you:

"So, what's new at 10.0?"

What kinds of calls are we likely to get?

What are the customers going to see?"

What do you say in response?

Summarize your answer for each of the above topics.

Summary and review, continued

- JFS (10.01)
- Kernel generation
- Diagnostics
- SAM
- Other administrative commands
- Virtual memory
- PRM
- Trusted systems

Notes

What else has changed?

- **NFS**
- **UUCP**
- **Curses**
- **VUE**
- **Patch program**
- **Termio**
- **Telnet**

Notes

- **The above topics are discussed in material appended to the end of this coursebook.**
- **This material was provided by various groups at HP.**
- **There is also a datasheet about a Kittyhawk training course.**
- **Additionally, refer to related manuals and the "Delta" document.**

What else has changed at
HP-UX 10.0?, continued



HEWLETT®
PACKARD

What else has changed? *Delta*

- STREAMS
- Language products
- libc
- Threads
- NLS

Delta
•
•
•
•
•
•
•
•

Notes

- The above topics are discussed in related manuals, and the "Delta" document.

Any questions?

Any questions?



Notes