1. Introduction

Purpose and Scope

This is the second version of the *AIX/HP-UX Interoperability Guide*, incorporating information on AIX 4 and HP-UX 10. The purpose of this document is to help users and system administrators of either Hewlett-Packard or IBM UNIX machines to understand the “other side,” if you will, for the purpose of interoperability. This will, for example, help AIX system administrators integrate HP-UX machines into their networks, or HP-UX administrators do the same with RS/6000s. It can also assist HP field representatives understand the nature of AIX in order to aid customers moving to HP machines.

The basic premise of this guide is that you can effectively integrate AIX and HP-UX machines without having to buy special products such as IBM’s NetView or HP’s Openview. An organization’s integration needs may very well require purchasing such products, but if you want just functional interoperability it is not necessary.

Also, this guide is in no way designed to be a complete handbook, one that can replace the extensive documentation provided by IBM and Hewlett-Packard. Both companies publish books that cover every aspect of system administration of their respective boxes. What the guide does is 1) assume you are already familiar with one or the other respective operating systems, AIX or HP-UX, 2) that all you need is a jump start to do key tasks with the operating system that are new to you, and 3) provide interoperability tips. Therefore each of the guide’s chapters covers basic system administration tasks and provides a brief discussion of how each operating system handles them and wherever possible concludes with an interoperability discussion.

When it comes to the topic of interoperability, there are a couple of things to keep in mind. First, this guide addresses system-level operability only and does not assume the use of any particular application. In other words, the tips contained in the guide have no particular end-user application in mind. Second, the issue of interoperability is not just how the two operating systems can work together, because outside of the network they don’t work together. Instead, they coexist. Or at least they should. So the guide simply helps an administrator understand just how things are done, where the files and directories are, how similarly named commands behave, what commands are equivalent in case their names differ. Ultimately we hope the book will help the administrator make peaceful coexistence as easy as possible.

The following are important parameters of this guide:

- Though the default shell for HP-UX 10 is now the POSIX shell, Korn shell syntax will be used for all the example commands, since the POSIX shell is similar to the Korn shell in many respects. The Korn shell remains the AIX’s default shell.
- The back slash (\) at the end of a command indicates line continuation because of insufficient space on the page. In real life the command can be typed on a single line.
- The SMIT examples often show “fastpath” commands. These are shortcuts to SMIT menu items and are optional. You can start at SMIT’s top-level menu and eventually get to the appropriate item.

- Additionally, the SMIT examples come from the ASCII version of SMIT rather than the Motif version. This made it easier to import SMIT menus into this document, but there is no difference in the menu items themselves.
- The machines used in the research for this guide were an IBM RS/6000 42T running AIX 4.1.4 and an HP 9000 712 running HP-UX 10.20.
- The RS/6000 was loaded with the Base Operating System (BOS) and AIXwindows (X11) licensed programs and a few optional (but not separately purchased) software packages, including InfoExplorer and the Common Desktop Environment (CDE).
- The HP 712 was loaded with the HP-UX 10 CDE Runtime Bundle.

HP-UX and AIX

Put simply, HP-UX is Hewlett-Packard's version of UNIX while AIX is IBM's UNIX. But of course there is much more to it than that. Since 1969 there have been many strains and variations of UNIX, but by the end of the 1970s two major forms of UNIX existed: System V, owned by AT&T and usually regarded as the descendant of the original UNIX, and BSD (Berkeley Software Distribution), a product of the Computer Systems Research Group at the University of California, Berkeley. Each had its special features as well as advocates, and for this reason workstation vendors, including HP and IBM, began creating their own versions of UNIX in the 1980s which more often than not contained elements of both AT&T and BSD types.

The HP-UX version 10 operating system is based on UNIX System V Release 4, with important features from the Fourth Berkely Software Distribution. It is essentially an AT&T-type of UNIX with numerous extensions. Its system administration tool is SAM (System Administration Manager). There are two major OS bundles in the 10.0 system: the Desktop bundle (Series 700 only) and the Runtime bundle. The HP-UX 10.0 Desktop bundle is a minimal "client" system. It does not include all of the products and filesets found in a fully complete Runtime bundle.

AIX version 4 is also based on UNIX System V and Berkeley Software Distribution 4.3 but is more of a hybrid of these two types of UNIX than HP-UX. It also contains several IBM-proprietary features, such as the Object Data Manager (ODM) and System Resource Controller (SRC). AIX 4 comes in three major packages: AIX 4.1 for Clients, AIX 4.1 for Servers, and AIX Connections Version 4.1, the latter providing PC-to-UNIX connectivity. AIX's system administration tool is SMIT (System Management Interface Tool).

2. System Startup and Shutdown

AIX

Normal Boot

If everything is configured properly, normal boot simply involves two things: 1) making sure the front panel key switch is in `NORMAL` position (all the way to the left), and 2) powering on the system. Everything else is automatic: the system comes up in multi-user mode, ready for someone to log in. The following sequence of events takes place when an RS/6000 is powered on or reset:

- ROS IPL (Read Only Storage Initial Program Load). This phase includes a power-on self-test (POST), the location of a boot device, and loading of the boot kernel into memory.
- Phase 1 (Base Device Configuration Phase). This phase runs **`/etc/rc.boot`** with an argument of 1. **`rc.boot`** builds the Object Data Manager (ODM) database, makes sure that base devices are configured, initializes the Logical Volume Manager (LVM), activates the root volume group (rootvg), and checks and mounts the root file system.
- Phase 2. Here **`/etc/rc.boot`** is run with an argument of 2. This merges the ODM data and device files into the root file system and configures any devices not configured by Phase 1.
- Phase 3. This phase starts **`/etc/init`** with the process id (pid) of 1.
- Phase 4 (Runtime phase). Here **`init`** runs the entries in **`/etc/inittab`** and invokes **`/etc/rc.boot 3`**. The **`/tmp`** file system is mounted, the ODM database is saved for future boots, and run state is set to multi-user, at which time various subsystems such as TCP/IP and NFS, if found in **`/etc/inittab`**, are started.

Below is a typical **`/etc/inittab`** file.

```
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console
      # Power Failure Detection
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
fbcheck:2:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # run
      /etc/firstboot
srcmstr:2:respawn:/usr/sbin/srcmstr # System Resource Controller
rctcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
cron:2:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1  # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
writesrv:2:wait:/usr/bin/startsrc -swritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd
logsymp:2:once:/usr/lib/ras/logsymptom # for system dumps
infod:2:once:startsrc -s infod
diagd:2:once:/usr/lpp/diagnostics/bin/diagd >/dev/console 2>&1
dt:2:wait:/etc/rc.dt
cons:0123456789:respawn:/usr/sbin/getty /dev/console
```

As you can see, **`/etc/inittab`** starts the Common Desktop Environment (CDE) in the **`lft`** via the **`/etc/rc.dt`** script. So instead of having what looks like an ASCII display providing a login prompt, you have something similar to **`xdm`** doing so. See the next chapter for more information about CDE.

Up until the runtime phase all you have as an indicator of how the boot sequence is going is the LED display on the front panel of the machine. Three-digit codes flash as the sequence progresses, and if you want to know the meaning of the codes, you have to look them up in the *Diagnostics Guide*. At a certain point, however, you will see either the code **`c32`** or **`c33`**, which indicates that the runtime phase is assigning the console. **`c32`** is for low-function terminal devices (**`lft`**s)[1] and **`c33`** is for serial-line terminals (**`tty`**s).

Single-user Boot

A single-user boot in AIX 4 requires nothing more than switching the key to **SERVICE** position and powering on the system. The first thing displayed is a **DIAGNOSTIC OPERATING INSTRUCTIONS** screen. Pressing **`Enter`** at this screen produces a **FUNCTION SELECTION** menu. Selection 5 is **`Single User Mode`**. Selecting that item produces a password prompt for the root account. Once you successfully enter the password, you enter single-user mode.

The diskette boot process found in AIX 3.2.5 does not exist in AIX 4.

System Shutdown

The following examples must be done while logged in as root.

To shut down the system in 10 minutes:

```
# shutdown +10
```

To do a fast shutdown now with no warnings:

```
# shutdown -F
# halt
```

To shut down with a warning for users to log off:

```
# shutdown now
```

To shut down with a one-minute warning and then reboot:

```
# shutdown -r
```

To reboot immediately:

```
# shutdown -Fr
# reboot
```

To shut down AIX to single-user mode:

```
# telinit S
# init s
# shutdown -m
```

[1]Low-fucntion terminals replace AIX version 3's high-function terminal (**hft**).

HP-UX

Normal Boot

The bootstrap process involves the execution of three software components:

- **`pdc`**
- **`isl`**
- **`hpux`**

pdc

Automatic boot processes on various HP-UX systems follow similar general sequences. When power is applied to the HP-UX system processor, or the system Reset button is pressed, the firmware processor-dependent code (**`pdc`**) is executed to verify hardware and general system integrity. After checking the hardware, **`pdc`** gives the user the option to override the autoboot sequence by pressing the **`Esc`** key. A message resembling the following usually appears on the console.

```
(c) Copyright. Hewlett-Packard Company. 1994.
All rights reserved.

PDC ROM rev. 130.0
32 MB of memory configured and tested.

Selecting a system to boot.
To stop selection process, press and hold the ESCAPE key...
```

If no keyboard activity is detected, **`pdc`** commences the autoboot sequence by loading **`isl`** and transferring control to it.

isl

The initial system loader (**`isl`**) implements the operating-system-independent portion of the bootstrap process. It is loaded and executed after self-test and initialization have completed successfully. Typically, when control is transferred to **`isl`**, an autoboot sequence takes place. An autoboot sequence allows a complete bootstrap operation to occur with no intervention from an operator. While an autoboot sequence occurs, **`isl`** finds and executes the autoexecute file which requests that **`hpux`** be run with appropriate arguments. Messages similar to the following are displayed by **`isl`** on the console:

```
Booting from: scsi.6  HP 2213A
Hard booted.
ISL Revision A.00.09  March 27, 1990
ISL booting  hpux boot disk(;0)/stand/vmunix
```

hpux

`hpux`, the secondary system loader, then announces the operation it is performing, in this case the boot operation, the device file from which the load image comes, and the TEXT size, DATA size, BSS size, and start address of the load image, as shown below, before control is passed to the image.

```
    Booting disk(scsi.6;0)/stand/vmunix
    966616+397312+409688 start 0x6c50
```

Finally, the loaded image displays numerous configuration and status messages, and passes control to the **`init`** process. At this point an HP-UX system resembles an AIX system in that **`init`** reads the **`/etc/inittab`** file to complete initialization. But there are also some significant differences, which are explained later in this chapter.

Single-user Boot

A single-user boot in HP-UX is sometimes referred to as an interactive boot or attended mode boot. Pressing the **`Escape`** key at the boot banner on an older Series 700 workstation halts the automatic boot sequence, puts you into attended mode, and displays the **Boot Console User Interface** main menu, a sample of which is below.

```
        Selecting a system to boot.
        To stop selection process, press and hold the ESCAPE key.

        Selection process stopped.

        Searching for Potential Boot Devices.
        To terminate search, press and hold the ESCAPE key.

        Device Selection    Device Path            Device Type
        -------------------------------------------------------------
        P0                  scsi.6.0               QUANTUM PD210S
        P1                  scsi.1.0               HP      2213A
        P2                  lan.ffffff-ffffff.f.f  hpfoobar

        b) Boot from specified device
        s) Search for bootable devices
        a) Enter Boot Administration mode
        x) Exit and continue boot sequence

        Select from menu:
```

In this case the system automatically searches the SCSI, LAN, and EISA interfaces for all potential boot devices—devices for which boot I/O code (IODC) exists. The key to booting to single-user mode is first to boot to ISL using the **`b)`** option. The ISL is the program that actually controls the loading of the operating system. To do this using the above as an example, you would type the following at the **Select from menu:** prompt:

```
 Select from menu: b p0 isl
```

This tells the system to boot to the ISL using the SCSI drive at address 6 (since the device path of P0 is **scsi.6.0**). After displaying a few messages, the system then produces the ISL> prompt.

Pressing the **`Escape`** key at the boot banner on newer Series 700 machines produces the Boot Administration Utility, as shown below.

```
   Command                            Description
   -------                            -----------
   Auto [boot|search] [on|off]        Display or set auto flag
   Boot [pri|alt|scsi.addr][isl]      Boot from primary, alt or SCSI
   Boot lan[.lan_addr][install][isl]  Boot from LAN
   Chassis [on|off]                   Enable chassis code
   Diagnostic [on|off]                Enable/disable diag boot mode
   Fastboot [on|off]                  Display or set fast boot flag
   Help                               Display the command menu
   Information                        Display system information
   LanAddress                         Display LAN station addresses
   Monitor [type]                     Select monitor type
   Path [pri|alt] [lan.id|SCSI.addr]  Change boot path
   Pim [hpmc|toc|lpmc]                Display PIM info
   Search [ipl] [scsi|lan [install]]  Display potential boot devices
   Secure [on|off]                    Display or set security mode
   -----------------------------------------------------------------
   BOOT_ADMIN>
```

To display bootable devices with this menu you have to execute the **`Search`** command at the BOOT_ADMIN> prompt:

```
 BOOT_ADMIN> search
 Searching for potential boot device.
 This may take several minutes.

 To discontinue, press ESCAPE.

    Device Path               Device Type
    --------------            ---------------
    scsi.6.0                  HP      C2247
    scsi.3.0                  HP      HP35450A
    scsi.2.0                  Toshiba CD-ROM

 BOOT_ADMIN>
```

To boot to ISL from the disk at device path **scsi.6.0** type the following:

```
 BOOT_ADMIN>boot scsi.6.0 isl
```

Once you get the ISL prompt you can run the **`hpux`** utility to boot the kernel to single-user mode:

```
 ISL>hpux -is
```

This essentially tells **`hpux`** to load the kernel (**`/stand/vmunix`**) into single-user mode (**`-is`**) off the SCSI disk drive containing the kernel. The **`-is`** option says to pass the string ***s*** to the **`init`** process (***i***), and the command **`init s`** puts the system in single-user mode. In fact, you will see something similar to the following after typing the above command:

```
Boot
: disk(scsi.6;0)/stand/vmunix
966616+397312+409688 start 0x6c50

   Kernel Startup Messages Omitted

INIT: Overriding default level with level ‘s’

INIT: SINGLE USER MODE
WARNING:  YOU ARE SUPERUSER!!
#
```

Startup

The startup and shutdown of various subsystems is an essential difference between AIX and HP-UX. For example, in AIX subsystems such as NFS and **`cron`** are started explicitly in **`/etc/inittab`**. Beginning with HP-UX 10 **`/etc/inittab`** calls **`/sbin/rc`**, which in turn calls execution scripts to start subsystems. This approach follows the OSF/1 industry standard and has been adopted by Sun, SGI, and other vendors. There are four components to this method of startup and shutdown: **`/sbin/rc`**, execution scripts, configuration variable scripts, and link files.

/sbin/rc

This script invokes execution scripts based on run levels. It is also known as the startup and shutdown sequencer script.

Execution scripts

These scripts start up and shut down various subsystems and are found in the **`/sbin/init.d`** directory. **`/sbin/rc`** invokes each execution script with one of four arguments, indicating the “mode”:

`start`	Bring the subsystem up
`start_msg`	Report what the start action will do
`stop`	Bring the subsystem down
`stop_msg`	Report what the stop action will do

These scripts are designed never to be modified. Instead, they are customized by sourcing in configuration files found in the **`/etc/rc.config.d`** directory. These configuration files contain variables that you can set. For example, in the configuration file **`/etc/rc.config.d/netconf`** you can specify routing tables by setting variables like these:

```
ROUTE_DESTINATION[0]="default"
ROUTE_GATEWAY[0]="gateway_address"
ROUTE_COUNT[0]="1"
```

The execution script **`/sbin/init.d/net`** sources these and other network-related variables when it runs upon system startup. More on configuration files is described below.

Upon startup a checklist similar to the one below will appear based upon the exit value of each of the execution scripts.

```
HP-UX Startup in progress
-----------------------------------
Mount file systems..............................[ OK ]
Setting hostname................................[ OK ]
Set privilege group.............................[ OK ]
Display date...................................[FAIL]*
Enable auxiliary swap space....................[ N/A ]
Start syncer daemon.............................[ OK ]
Configure LAN interfaces........................[ OK ]
Start Software Distributor agent daemo..........[ OK ]
```

The execution scripts have the following exit values:

0 Script exited without error. This causes the status **OK** to appear in the checklist.

1 Script encountered errors. This causes the status **FAIL** to appear in the checklist.

2 Script was skipped due to overriding control variables from **`/etc/rc.config.d`** files or for other reasons, and did not actually do anything. This causes the status **N/A** to appear in the checklist.

3 Script executed normally and requires an immediate system reboot for the changes to take effect. (NOTE: Reserved for key system components).

Configuration variable scripts

Configuration variable scripts are designed to customize the execution scripts. This goal here is to separate startup files from configuration files so that upgrading your system does not overwrite its configuration. These scripts are written for the POSIX shell (**`/usr/bin/sh`** or **`/sbin/sh`**), and not the Bourne shell, **`ksh`**, or **`csh`**. In some cases, these files must also be read, and possibly modified by other scripts or the SAM program. For this reason, each variable definition must appear on a separate line, in the syntax:

```
variable=value
```

No trailing comments may appear on a variable definition line. Comment statements must be on separate lines, with the “#” comment character in column 1. An example of the required syntax for configuration files is given below:

```
# Cron configuration. See cron(1m)
#
# CRON: Set to 1 to start cron daemon
#
CRON=1
```

Both the execution scripts and the configuration files are named after the subsystem they control. For example, the **`/sbin/init.d/cron`** execution script controls the **`cron`** daemon, and it is customized by the **`/etc/rc.config.d/cron`** configuration variable script.

Link Files

These files control the order in which execution scripts run. The **`/sbin/rc#.d`** (where **`#`** is a run-level) directories are startup and shutdown sequencer directories. They contain only symbolic links to the execution scripts in **`/sbin/init.d`** that are executed by **`/sbin/rc`** on transition to a specific run level. For example, the **/sbin/rc3.d** directory contains symbolic links to scripts that are executed when entering run level 3.

These directories contain two types of link files: start links and kill links. Start links have names beginning with the capital letter **S** and are invoked with the **start** argument at system boot time or on transition to a higher run level. Kill links have names beginning with the capital letter **K** and are invoked with the **stop** argument at system shutdown time, or when moving to a lower run level.

Further, all link files in a sequencer directory are numbered to ensure a particular execution sequence. Each script has, as part of its name, a three-digit sequence number. This, in combination with the **start** and **kill** notation, provides all the information necessary to properly start up and shut down a system.

The table below shows some samples from the run-level directories. (The sequence numbers shown are only for example and may not accurately represent your system.)

/sbin/rc0.d	/sbin/rc1.d	/sbin/rc2.d	/sbinrc3.d
K480syncer	S100hfsmount	S340net	S000nfs.server
K800killall	S320hostname	S500inetd	
K900hfsmount	S440savecore	S540sendmail	
	S500swapstart	S610rbootd	
	S520syncer	S720lp	
		S730cron	
	K270cron		
	K280lp	K900nfs.server	
	K390rbootd		
	K460sendmail		
	K500inetd		
	K660net		

Because each script in **/sbin/init.d** performs both the startup and shutdown functions, each will have two links pointing towards the script from **/sbin/rc*.d**; one for the **start** action and one for the **stop** action.

Run Levels and /sbin/rc

In previous HP-UX releases, **/etc/rc** (now **/sbin/rc**) was run only once. Now it may run several times during the execution of a system, sequencing the execution scripts when moving between run levels. However, only the subsystems configured for execution, through configuration variables in **/etc/rc.config.d**, are started or stopped when transitioning the run levels.

/sbin/rc sequences the startup and shutdown scripts in the appropriate sequencer directories in lexicographical order. Upon transition from a lower to a higher run level, the **start** scripts for the new run level and all intermediate levels between the old and new level are executed. Upon transition from a higher to a lower run level, the **kill** scripts for the new run level and all intermediate levels between the old and new level are executed.

When a system is booted to a particular run level, it will execute startup scripts for all run levels up to and including the specified level (except run level 0). For example, if booting to run level 4, **/sbin/rc** looks at the old run level (S) and the new run level (4) and executes all start scripts in states 1, 2, 3, and 4. Within each level, the **start** scripts are sorted lexicographically and executed in that order. Each level is sorted and executed separately to ensure that the lower level subsystems are started before the higher level subsystems.

Consequently, when shutting down a system, the reverse takes place. The **`kill`** scripts are executed in lexicographical order starting at the highest run level and working down, as to stop the subsystems in the reverse order they were started. As mentioned earlier, the numbering is reversed from the startup order.

Example

If you want **`cron`** to start when entering run level 2, you would modify the configuration variable script **`/etc/rc.config.d/cron`** to read as follows:

```
# cron config
#
# CRON=1 to start

CRON=1
```

This would be necessary because the execution script, **`/sbin/init.d/cron`** contains the following:

```
# cron startup
#
. /etc/rc/config

if [ $CRON = 1 ]
   then /usr/sbin/cron
fi
```

`cron` will start at run level 2 because in **`/sbin/rc2.d`** a link exists from **`S730cron`** to **`/sbin/init.d/cron`**. **`/sbin/rc`** will invoke **`/sbin/init.d/cron`** with a **`start`** argument because the link name starts with an **`S`**.

System Shutdown

To shut down HP-UX for power-off, you can do any of the following:

```
# init 0
# shutdown -h -y now
```

To shut down and reboot HP-UX:

```
# reboot
# shutdown -r -y now
```

To shut down HP-UX to single-user mode:

```
# init S
# shutdown -y now
# shutdown 0
```

The **`-h`** option to the **`shutdown`** command halts the system completely but will prompt you for a message to issue users. The **`-y`** option completes the shutdown without asking you any of the questions it would normally ask.

Summary

The starting and stopping of subsystems differs dramatically between AIX and HP-UX. Starting subsystems at bootup in AIX is done explicitly via the **`/etc/inittab`** file, and the **`shutdown`** command stops all subsystems. In HP-UX subsystems are started by execution scripts via the **`/sbin/rc`** script only if they are configured to do so by configuration variable scripts. They are also started in a predefined order according to run level. Moreover, the stopping of subsystems is also governed by a sequencing scheme based on run level. This mechanism allows for administrators to fine-tune the startup and shutdown of subsystems and replaces the older **`/etc/shutdown.d`** system found in HP-UX version 9.

3. User Environment and Login

AIX

If you install the Base Operating System (BOS) of AIX on an RS/6000 with a graphics adapter attached to the console, then the following software packages are automatically installed:

- **`bos.rte`** (Base Operating System Runtime)
- **`bos`** (Base Operating System)
- **`X11`** (AIXwindows)

Your login environment at this stage is a **`getty`** process running on a low-function terminal (**`lft`**), a device that emulates a character device on a graphics display. Logging into this environment provides you with what can now be called an "old-fashioned" command-line interface, essentially the KornShell (**`ksh`**) with a **`$`** prompt. Entering the following command will start an AIXwindows interface:

```
$ xinit
```

However, the primary graphical user interface found on AIX systems today is now the Common Desktop Environment (CDE). This interface is an industry standard and is also available on HP-UX beginning with version 10.10. CDE is not automatically installed in a BOS installation, so you have to load the **`X11.Dt`** product in order to use it. Doing so not only installs CDE but makes it the default environment upon bootup. To disable CDE for subsequent boots, type the following at the command line:

```
$ dtconfig -d
```

To enable CDE for subsequent boots, type the following:

```
$ dtconfig -e
```

CDE Startup and Login

As mentioned in the previous chapter, once a machine has booted, the **`init`** process reads **`/etc/inittab`**. This file calls the **`/etc/rc.dt`** script, which in turn calls **`/usr/dt/bin/dtlogin`**, the login server and display manager.

The following is an abridged version of the **`dtlogin`** process:

1. **`init`** starts **`dtlogin`** upon bootup.
2. **`dtlogin`** starts the X server for the local display, if necessary.
3. **`dtlogin`** starts a new **`dtlogin`** process for each display it manages.
4. The new **`dtlogin`** process starts **`dtgreet`**, which displays the login screen and handles the user's interaction with the login screen.
5. When a user logs in to the **`dtgreet`** display, **`dtlogin`** runs **`/usr/dt/config/Xstartup`**, if it exists, sets certain environment variables to default values, runs **`/usr/dt/bin/Xsession`**, which in turn reads **`$HOME/.dtprofile`** and invokes the session manager, **`/usr/dt/bin/session`**.

6. **`dtlogin`** handles the login process instead of the **`login`** process found on traditional UNIX systems.

The files that affect a user's environment differ, depending on whether the user interface one is using. The table below lists those interfaces and the some of the files associated with them:

Command line	AIXwindows	CDE
/etc/environment	/etc/environment	/etc/environment
/etc/profile	/etc/profile	$HOME/.dtprofile
$HOME/.profile	$HOME/.profile	$HOME/.dt/dtwmrc
	$HOME/.Xdefaults	
	$HOME/.xinitrc	
	$HOME/.mwmrc	

`/etc/environment` contains default variables for each process created by the **`exec()`** system call and affects all three environments. Variables include PATH, TZ, and LANG. **`/etc/profile`** contains environment variables and commands that are invoked each time a user logs in. This file affects only the command line and AIXwindows environments. For the Common Desktop Environment, **`$HOME/.dtprofile`** and the files in the **`$HOME/.dt`** directory are important configuration files. Normally, a user's **`$HOME/.profile`** file is not read in CDE unless the `DTSOURCEPROFILE` variable is set to "true", usually in **`$HOME/.dtprofile`**. Configuring CDE is more complex than editing these files. For in-depth information consult the *Common Desktop Environment: Advanced User's and System Administrator's Guide*, available in InfoExplorer.

The **`Options`** menu on the CDE login screen allows you to select an alternative type of session: a fail-safe session. You can also select the language for your session. To log into and out of a fail-safe session, select **`Session`** from the **`Options`** menu and then choose **`Failsafe Session`** to log in. A fail-safe session is a simple session that starts **`Motif`** and an **`aixterm`** window.

You can log into a non-desktop environment by selecting **`Command Line Login`** from the **`Options`** menu. Logging in this way temporarily terminates the X server and provides access to the system via the **`lft`**. Logging out of the **`lft`** (typing **`exit`** at the command line) restarts the X server.

Logout

To log out of the destkop, choose **`Log out . . .`** from the **`Workspace Menu`** or click on the **`Exit`** control located next to the workspace buttons in the Front Panel. To log out of an **`lft`** session, type **`exit`** or **`logout`** at the command line.

User Attributes

AIX maintains several databases containing user information that the system administrator can alter to fine-tune security and meet the requirements of his or her local site. These databases take the form of ASCII files containing stanzas of information for each user. Stanzas labeled "default" contain values used by the **`mkuser`** command when creating a new user. For example, when using SMIT to create a user (fastpath **`smit mkuser`**) you see a list of the attributes that can be set:

```
                              Add a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                     [Entry Fields]
* User NAME                                            []
  User ID                                              []              #
  ADMINISTRATIVE USER?                                  false          +
  Primary GROUP                                        []              +
  Group SET                                            []              +
  ADMINISTRATIVE GROUPS                                []              +
  Another user can SU TO USER?                          true           +
  SU GROUPS                                            [ALL]           +
  HOME directory                                       []
  Initial PROGRAM                                      []
  User INFORMATION                                     []
  EXPIRATION date (MMDDhhmmyy)                         [0]
  Is this user ACCOUNT LOCKED?                          false          +
  User can LOGIN?                                       true           +
  User can LOGIN REMOTELY?                              true           +
  Allowed LOGIN TIMES                                  []
  Number of FAILED LOGINS before                       [0]             #
       user account is locked
  Login AUTHENTICATION GRAMMAR                         [compat]
  Valid TTYs                                           [ALL]
  Days to WARN USER before password expires            [0]             #
  Password CHECK METHODS                               []
  Password DICTIONARY FILES                            []
  NUMBER OF PASSWORDS before reuse                     [0]             #
  WEEKS before password reuse                          [0]             #
  Weeks between password EXPIRATION and LOCKOUT        [-1]
  Password MIN. AGE                                    [0]             #
  Password MIN. LENGTH                                 [0]             #
  Password MIN. ALPHA characters                       [0]             #
  Password MIN. OTHER characters                       [0]             #
  Password MAX. REPEATED characters                    [8]             #
  Password MIN. DIFFERENT characters                   [0]             #
  Password REGISTRY                                    []
  MAX. FILE size                                       [2097151]       #
  MAX. CPU time                                        [-1]
  MAX. DATA segment                                    [262144]        #
  MAX. STACK size                                      [65536]         #
  MAX. CORE file size                                  [2048]          #
  File creation UMASK                                  [022]
  AUDIT classes                                        []              +
  TRUSTED PATH?                                         nosak          +
  PRIMARY authentication method                        [SYSTEM]
  SECONDARY authentication method                      [NONE]
```

The attributes listed above are stored in a series of files:

`/etc/passwd`	Contains the basic attributes of users
`/etc/group`	Contains the basic attributes of groups
`/etc/security/user`	Contains the extended attributes of users

/etc/security/limits	Contains the process resource limits of users
/etc/security/laslog	Contains the last login attributes for users
/usr/lib/security/mkuser.default	Contains the default values for user accounts

You can use SMIT to view or change user attributes, or you can use the **lsuser** or **chuser** commands. You can add or remove a user using SMIT or the **mkuser** and **rmuser** commands. For example, to list the attributes of the user **nan** in stanza format:

```
# lsuser -f nan
nan:
        id=201
        pgrp=roberts
        groups=roberts,staff
        home=/home/nan
        shell=/usr/bin/ksh
        login=false
        su=false
        rlogin=true
        daemon=true
        admin=false
        sugroups=ALL
        admgroups=
        tpath=nosak
        ttys=ALL
        expires=0
        auth1=SYSTEM
        auth2=NONE
        umask=22
        SYSTEM=compat
        logintimes=
        loginretries=0
        pwdwarntime=0
        account_locked=false
        minage=0
        maxage=0
        maxexpired=-1
        minalpha=0
        minother=0
        mindiff=0
        maxrepeats=8
        minlen=0
        histexpire=0
        histsize=0
        pwdchecks=
        dictionlist=
        fsize=2097151
        cpu=-1
        data=262144
        stack=65536
        core=4096
        rss=65536
        time_last_login=839042643
        tty_last_login=/dev/pts/0
        host_last_login=hpubvwa.nsr.hp.c
        unsuccessful_login_count=0
```

To change **nan**'s SYSTEM attribute from **compat** to **NONE**:

```
# chuser SYSTEM=NONE nan
```

To see the change made by the previous command by displaying only the `SYSTEM` attribute:

```
# lsuser -a SYSTEM nan
nan SYSTEM=NONE
```

Normally a user is authenticated by two files: **`/etc/passwd`**, which verifies the account, and **`/etc/security/passwd`**, which contains the encrypted passwords and other security-related data. An important user attribute in user authentication is the `SYSTEM` attribute. This attribute may be set to one of the following:

- **DCE -** Users authenticated by means of the Distributed Computing Environment
- **compat -** Users authenticated by the local password files first and then, if necessary, by Network Information Services (NIS) second.
- **files -** Users authenticated by local files only.
- **NONE -** No authentication takes place.

Note that a value of `NONE` for the `SYSTEM` attribute will turn off authentication for a user, *even if there is still an encrypted password for that user in* ***`/etc/security/passwd`****.*

HP-UX 10

VUE

The process described below is more accurately described as the HP VUE login process. VUE is HP's Visual User Environment, a graphical user interface for HP-UX. If you were to login to a non-Windows console or were to **`telnet`** from another machine to an HP-UX system, the login process would very much be the same as that of a typical UNIX system, which involves the **`init`**, **`getty`**, and **`login`** programs. However, the **`/etc/inittab`** file of most HP machines has HP VUE as part of the default run level; in other words, HP VUE starts by default. HP VUE's login program, **`vuelogin`** (which is a customized version of **`xdm`**), provides the same functions as **`init`**, **`getty`**, and **`login`**, and does not normally read such files as **`/etc/profile`** and **`.profile`**.

An abridged version of the **`vuelogin`** process is as follows:

1. **`init`** starts **`vuelogin`** upon bootup.
2. **`vuelogin`** starts the X server.
3. **`vuelogin`** starts a new **`vuelogin`** process for each display it manages.
4. The new **`vuelogin`** invokes **`vuegreet`**, which displays the login screen and handles the user's interaction with the login screen.
5. When a user logs in to the **`vuegreet`** display, **`vuelogin`** runs **`/usr/vue/config/Xstartup`** if it exists, sets certain environment variables to default values, runs **`/usr/vue/config/Xsession`**, which in turn reads **`$HOME/.vueprofile`** and invokes the session manager, **`vuesession`**.
6. Normally, a user's **`$HOME/.profile`** file is not read unless you uncomment the following line in **`$HOME/.vueprofile`**:

```
# VUE=true: export VUE; . $HOME/.profile; unset VUE # sh, ksh
```

Alternative Login Methods

The **`Options`** menu on the HP VUE login screen allows you to select several alternative types of sessions, such as HP VUE Lite, or a fail-safe session. You can also select the language for your session.

If you choose not to use HP VUE, you can select **`Options`** from the login window menu and **`No Windows`** from the VUE login screen, at this time. In that case, enter your login name and password after the appropriate prompts.

To log into and out of a fail-safe session, select **`Fail-safe`** from the **`Options`** menu and log in. A fail-safe session is a simple session that starts the Workspace Manager and a single terminal window. It is useful when you need access to a single Terminal Emulator window to execute several commands before logging into an HP VUE session.

Logout

To log out if you are in an HP VUE session, choose the logout control on the Front Panel, use the log out control to end the session, or choose Log out from the workspace menu. When you log out of a regular HP VUE session, Session Manager saves information about your current session so that it can be restored the next time you log in.

CDE on HP-UX

As stated earlier, beginning with HP-UX 10.10 users have a choice of which user interface to use: either HP VUE or CDE. The Common Desktop Environment is nearly the same on both AIX and HP-UX, but there is one notable exception. In AIX CDE starts by a direct call in to **`/etc/rc.dt`** by **`/etc/inittab`**. In HP-UX **`/etc/inittab`** calls **`/sbin/rc`**, which starts the execution script **`dt`** when it enters run-level 3 (see Chapter 2).

A Comparison of HP VUE and CDE

The following tables summarize the differences between VUE and CDE.

User Differences Between HP VUE and CDE

Component	**HP VUE**	**CDE**
File Manager	Actions menu contains actions for selected file	Selected menu contains actions for selected file. Tree view preference is enhanced.
Style Manager	--	Screen dialog box provides screen savers.
Front Panel	--	Front Panel can be customized using pop-up menus and the subpanel's Install Icon control.
Help Manager	Help volume and index for each help application	Index is global for all help volumes.
Application Container(s)	Personal, General, and Network Toolboxes	Application Manager

Calendar	Not available	Sets appointments. Calendar information can be shared amongst multiple users.
Mailer	Default is elm.	Graphical mailer dtmail that accepts attachments.
Text editor	Default is HP VUE Text Editor (vuepad).	Default is CDE Text Editor (dtpad).
Terminal emulator	Default is hpterm.	Default is dtterm.
Login Manager	Graphical login screen with login options.	One text box for name and password.
Session Manager	Current or home session restored. VUE-Lite sessions available.	Current or home session restored. No Lite sessions.
Print Manager	Not available.	Displays icon for each local or network printer. Icons are used to print files and obtain printer status.
File printing	Can print a file by dropping it on a Front Panel printer control.	Can print a file by dropping it on a Front Panel Print control or a printer icon in Print Manager. Drag and drop to printer controls always displays a Print dialog box.
Trash Can	Lists deleted files.	Deleted files are represented as icons. You can drag icons from File Manager to the Trash Can window.
Workspaces	Default number is six.	Default number is four. You add workspaces using a pop-up menu.
File Annotater	File Annotater application.	Not available.
Icon creation and editing.	Icon Editor (vueicon).	Icon Editor (dticon).
Calculator	Not provided.	CDE Calculator application.

Component	**HP VUE**	**CDE**
Personal environment	.vueprofile (.profile not automatically read).	.dtprofile (.profile not automatically read).
System-wide environment	/usr/vue/config/Xsession	/usr/dt/config/Xsession.d/*
Personal applications	Personal Toolbox	Personal application groups located on application search path.
System-wide applications	General Toolbox	System-wide application groups located on applications search path.
Networked applications	Network Toolbox	Application groups from CDE application servers automatically added to application Manager.
Application integration tools	Not provided.	dtappintegrate. Allows you to store desktop configuration files for an application under one location.
Action definition files		Uses new syntax and naming convention (*.dt). Files must be con CDE database search path.
Filetype (data type) definitions		Uses new syntax and naming convention (*.dt). Files must be on CDE database search path.
Create Action application	Creates action definitions.	Creates action and data type definitions.
Application preferences	For mailer, terminal, editor, and printing. In user-prefs.vf files.	For editor, terminal, and trash. In user-prefs.dt file.
Remote application invocation	Uses subprocess control daemon (spcd).	Uses subprocess control daemon (dtspcd) and ToolTalk filename databse server daemon (rpc.ttdbserver).
Remote application authorization	Automatic xhosting.	X mechanism ($HOME/.Xauthority)
Inter-Application Messaging System	Broadcase Message Server (BMS)	TookTalk
Mount points	--	Assumes /net. Can be changed using DTMOUNTPOINT variable.
Remote data	Could use hostname:/path syntax.	Must use file system name (for example, /nfs/hostname/path).
Mail infrastructure	--	Desktop mailer uses

		sendmail.
Remote printer access	--	Print Manager uses lp print spooler.
Front Panel customization	Definitions in vuewmrc or sys.vuewmrc	New syntax. Uses all files name *.fp on database search path.
Icons	Bitmaps and pixmaps in different directories. Files must be on VUE icon search path. File Manager uses “large” and “small” icons.	Bitmaps and pixmaps in same directories. Files must be on CDE icon search path. File Manager uses “medium” and “tiny” icons.
Color palettes	Style Manager lists personal and built-in palettes.	Syle Manager lists personal, system-wide, and built-in paletees. Palett colors used differently.
Color usage	Default uses eight color sets for high-color displays	Default uses four color sets for high-color displays.
Backdrops	Style Manager lists backdrops in one specified location.	Style Manager lists backdrops in multiple specified locations.
Fonts	--	CDE generic font names assigned to specific vendor fonts.
Application resources	Changed using xrdb.	Session Manager reads personal and system-wide resource files.
Terminal customization	hpterm is a Term0-compatible terminal that uses HP-specific escape sequences.	dtterm is DEC VT220-compatible terminal that uses ANSI escape sequences.

General File Locations

Functionality	**HP VUE Location**	**CDE Location**
Installed (Built-in) files	/usr/vue/*	/usr/dt/*
System-wide customizations	/etc/vue/*	/etc/dt/*
Personal cusomizations	$HOME/.vue/*	$HOME/.dt/*

Locations of Specific System-Wide Components

Functionality	HP VUE Location	CDE Location
Desktop application binaries	/usr/vue/bin	/usr/dt/bin
Desktop application app-defaults	/usr/vue/app-defaults	/usr/dt/app-defaults
Login Manager built-in files	/usr/vue/config/*	/usr/dt/config/*
Login Manager system-wide customization	/etc/vue/config/*	/etc/dt/config/*
Login Manager resources	/usr/vue/config/Xresources	/etc/dt/config/$LANG/Xresources
Session customization script	$HOME/.vueprofile	$HOME/.dtprofile
System-wide customization scripts	/etc/vue/config/Xsession /etc/vue/config/Xession.d/*	/etc/dt/config/Xsession.d/*
Built-initial session	/usr/vue/config/sys.*	/usr/dt/config/$LANG/sys.*
System-wide initial session	/etc/vue/config/sys.*	/etc/dt/config/$LANG/sys.*
Saved sessions	$HOME/.vue/sessions	$HOME/.dt/sessions
Display-dependent	$HOME/.vue/display/*	$HOME/.dt/display/*
Lite sessions	$HOME/.vue/sessions/lite/*	Not available
Session error log	$HOME/.vue/errorlog	$HOME/.dt/errorlog
Built-in Icons	/usr/vue/icons/*	/usr/dt/appconfig/icons/$LANG
System-wide icons	/etc/vue/icons/*	/etc/dt/appconfig/icons/$LANG
Personal icons	$HOME/.vue/icons/*	$HOME/.dt/icons
Built-in ac, data types	/usr/vue/types	/usr/dt/appconfig/types/$LANG
System-wide action	/usr/vue/config/types	/etc/dt/appconfig/types/$LANG
Personal actions, data types	$HOME/.vue/types	$HOME/.dt/types
Exported actions, data types	/etc/vue/config/export	/etc/dt/appconfig/types/$LANG
Application preferences	/etc/vue/types/user-prefs.vf	/usr/dt/appconfig/types/$LANG/user-prefs.dt

Application groups	Not available	/usr/dt/appconfig/appmanager/$LANG /etc/dt/appconfig/appmanagre/$LANG $HOME/.dt/appmanager/$LANG
Built-in help volume files	/usr/vue/help/$LANG/*	/usr/dt/appconfig/help/$LANG
System-wide help volume files	--	/etc/dt/appconfig/help/$LANG
Personal help volume files	--	$HOME/.dt/help
Built-in Front Panel definition	/usr/vue/config/sys.vuewmrc	/usr/dt/appconfig/types/$LANG/dtwm.fp
System-wide Front Panel definition	/etc/vue/config/sys.vuewmrc	/etc/dt/appconfig/types/$LANG/*.fp
Personal Front Panel definition	$HOME/.vue/vuewmrc	$HOME/.dt/types/*.fp
Built-in window manager configuration	/usr/vue/config/sys.vuewmrc	/usr/dt/config/$LANG/sys.dtwmrc
System-wide window manager configuration	/etc/vue/config/sys.vuewmrc	/etc/dt/config/$LANG/sys.dtwmrc
Personal window manager configuration	$HOME/.vue/vuewmrc	$HOME/.dt/dtwmrc

Summary

Logging into AIX and HP-UX systems is pretty much the same, now that the Common Desktop Environment has arrived. The biggest difference between the two with regard to users centers around AIX's system of user attributes and the use of **`/etc/security/passwd`**. Also, the default login shell is **`/usr/ksh`** on AIX and **`/usr/bin/sh`** (POSIX shell) on HP-UX. Also, in HP-UX's SAM program there is the ability to create NIS users as well as local users.

4. Devices

AIX

Introduction

AIX manages devices in a hierarchical fashion, where devices are grouped according to common characteristics. At the top of the hierarchy are **`functional classes`**, devices that perform the same basic function. For example, adapters of all types are part of the **`adapter`** functional class. Standard input/output adapters and microchannel adapters are examples of devices that belong to the next class, **`functional subclasses`**, devices that can be grouped more specifically according to function. Standard input/output adapters belong to the **`sio`** functional subclass, and microchannel adapters belong to the **`mca`** functional subclass. Finally, you have the **`device type`** class, devices that can be identified by model and manufacturer. For example, within the **`sio`** functional subclass are **`8fba`** (standard SCSI I/O controller), **`fda_2`** (standard I/O diskette adpater), and **`ient_6`** (integrated Ethernet adapter) device types, among others. Within the **`mca`** functional sublcass can be found the **`sio_2`** (standard I/O planar) device type. The command below illustrates this device hierarchy:

```
# lsdev -C -F "class subclass type description" | sort
adapter        buc        4006       GXT150L Graphics Adapter
adapter        mca        sio_2      Standard I/O Planar
adapter        sio        8fba       Standard SCSI I/O Controller
adapter        sio        fda_2      Standard I/O Diskette Adapter
adapter        sio        ient_6     Integrated Ethernet Adapter
adapter        sio        keyboard_2 Keyboard Adapter
adapter        sio        mouse      Mouse Adapter
adapter        sio        ppa        Standard I/O Parallel Port Adapter
adapter        sio        s1a        Standard I/O Serial Port 1
adapter        sio        s2a        Standard I/O Serial Port 2
adapter        sio        tablet_2   Tablet Adapter
aio            node       aio        Asynchronous I/O
bus            sys        mca        Microchannel Bus
cdrom          scsi       scsd       SCSI Multimedia CD-ROM Drive
disk           scsi       2000mb     2.0 GB SCSI Disk Drive
diskette       siofd      fd         Diskette Drive
if             EN         en         Standard Ethernet Network Interface
if             EN         ie3        IEEE 802.3 Ethernet Network Interface
if             LO         lo         Loopback Network Interface
ioplanar       sys        ioplanar_2 I/O Planar
keyboard       std_k      kb101      United States keyboard
lft            node       lft        Low Function Terminal Subsystem
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume lvsubclass lvtype     Logical volume
logical_volume vgsubclass vgtype     Volume group
lvm            lvm        lvdd       LVM Device Driver
memory         sys        simm       32 MB Memory SIMM
memory         sys        simm       32 MB Memory SIMM
memory         sys        simm       32 MB Memory SIMM
memory         sys        simm       32 MB Memory SIMM
mouse          std_m      mse_3b     3 button mouse
planar         sys        sysplanar3 System Planar
processor      sys        proc1      Processor
```

```
pty             pty         pty         Asynchronous Pseudo-Terminal
rcm             node        rcm         Rendering Context Manager Subsystem
sys             node        sys1        System Object
sysunit         sys         sysunit     System Unit
tape            scsi        8mm5gb      5.0 GB 8mm Tape Drive
tcpip           TCPIP       inet        Internet Network Extension
```

Lower levels have dependencies on higher levels. For example, the **`8mm5gb`** device type depends on the configuration of the **`scsi`** functional sublcass. The hierarchies and dependencies of devices in AIX are maintained by Object Data Manager (ODM) databases.

Object Data Manager

The ODM manages object-oriented databases of system information. Among the databases it manages is that of device configuration. This database is made up of two object classes: the **`predefined`** object class and the **`customized`** object class. The predefined object class contains configuration information for all possible devices supported by AIX, and the customized object class contains information for devices actually on the system.

Object Classes

An **`object class`** is a collection of objects with the same definition. Object class definitions resemble C language structures and are found in the **`/etc/objrepos`** (short for object repository) directory or in the directory specified by the `ODMDIR` variable. For example, the object definition for customized devices attached to the system is found in **`/etc/objrepos/CudDv`** and can be obtained by typing the **`odmshow`** command.

```
# odmshow  CuDv
class CuDv {
        char name[16];
        short status;
        short chgstatus;
        char ddins[16];
        char location[16];
        char parent[16];
        char connwhere[16];
        link PdDv PdDv uniquetype PdDvLn[48];
        };
```

Each of the items in the **`CuDv`** object class definition is a descriptor. When an object is added to the **`CuDv`** object class, values are associated with each of the descriptors. For example, a SCSI hard disk with a location code of `00-00-0S-00` would have this code stored in the **`location`** descriptor.

Objects

An **`object`** is an item that belongs to an object class. Each object has a set of configuration parameters corresponding to the descriptors in the object class definition. For example, to see the objects and their associated values that make up the **`CuDv`** object class, type:

```
#  odmget CuDv
```

Among the stanzas in the output, you would likely see the following:

```
CuDv:
     name = "hdisk0"
     status = "1"
     chgstatus = "2"
```

```
        ddins = "scdisk"
        location = "00-00-0S-00"
        parent = "scsi0"
        connwhere = "00"
        PdDvLn = "disk/scsi/1000mb"
```

Each of the above lines corresponds to the **`CuDv`** object class definition. The last line is important in that it illustrates the hierarchical classification of devices. As stated earlier, the ODM database consists of more than device data, but within the realm of devices there is similar "class" terminology, which can be confusing at times. In the last line above, for example, **`disk`** is the functional class, **`scsi`** is the functional subclass, and **`1000mb`** is the device type.

Adding Supported Devices

Adding and removing devices requires manipulating the device configuration database via the ODM, a daunting task. If you use AIX device management commands like **`mkdev`**, **`rmdev`**, **`lsdev`**, **`lsattr`**, and **`chdev`** you need to be familiar with these class concepts and the ODM. However, two tools can greatly simplify the task: the **`cfgmgr`** command and SMIT.

cfgmgr

The **`cfgmgr`** command, also known as the Configuration Manager, automatically configures devices on the system. It runs twice at system boot and can be run at the command line as well. When invoked, **`cfgmgr`** reads rules from the **`Config_Rules`** object class. These rules are commands that configure devices and make them available for use by creating device files and installing drivers. To see what these rules are, use the **`odmget Config_Rules`** command.

Thus one way of adding, say, an 8mm, 5GB SCSI tape drive is to:

1. Power down the system
2. Connect the drive
3. Power-on the drive
4. Boot the system

During system boot, **`cfgmgr`** will recognize the new device and configure it for use. If you type the following command, you will see that the drive is recognized:

```
 # lsdev -C -c tape
 rmt0 Available 00-00-0S-1,0 5.0 GB 8mm Tape Drive
```

The term "Available" refers to the tape drive's state. A device has one of three states: **`undefined`**, **`defined`**, or **`available`**. If the tape drive is **`undefined`**, it has not been configured by **`cfgmgr`**. If it is **`defined`**, its configuration data is recorded in the customized database, but the device is not available for use. Once a defined device is bound to the kernel it is in the **`available`** state. **`cfgmgr`** will make sure your device is defined and then made available.

`cfgmgr` can also be run at the command line. If you add a device, such as the above tape drive, to a running system (even if it is a SCSI device), you have to run the **`cfgmgr`** command to make it available. Also, if you have an available but powered-down device when you reboot, the **`cfgmgr`** during bootup will set the device to a **`defined`** state. To make it available, power-on the device and rerun the **`cfgmgr`** command.

Adding a Device Using SMIT

You can attach a device to the system, power it on, and then use SMIT to configure it. To do so, just type **`smit device`** at the command line. You will see the following:

```
                                 Devices

Move cursor to desired item and press Enter.

  Install/Configure Devices Added After IPL
  Printer/Plotter
  TTY
  Asynchronous Adapters
  PTY
  Console
  Fixed Disk
  Disk Array
  CD ROM Drive
  Read/Write Optical Drive
  Diskette Drive
  Tape Drive
  Communication
  Graphic Displays
  Graphic Input Devices
  Disk Array
  CD ROM Drive
  Read/Write Optical Drive
  Diskette Drive
  Tape Drive
  Communication
  Graphic Displays
  Graphic Input Devices
  Low Function Terminal (LFT)
  SCSI Initiator Device
  SCSI Adapter
  Asynchronous I/O
  Multimedia
  List Devices
  Install Additional Device Software
```

Just select the item you wish to add and press **`Enter`**. If you were to add the example 8mm tape drive to the system this way, you would select **`Tape Drive`**. The next screen would look like the following:

```
                               Tape Drive

Move cursor to desired item and press Enter.

  List All Defined Tape Drives
  List All Supported Tape Drives
  Add a Tape Drive
  Change / Show Characteristics of a Tape Drive
  Remove a Tape Drive
  Configure a Defined Tape Drive
  Generate Error Report
  Trace a Tape Drive
```

Select **`Add a Tape Drive`**. You will then be presented a list of supported tape drives, something similar to the following:

```
                          Tape Drive Type

   Move cursor to desired item and press Enter.

     1200mb-c scsi 1.2 GB 1/4-Inch Tape Drive
     150mb    scsi 150 MB 1/4-Inch Tape Drive
     3490e    scsi 3490E Autoloading Tape Drive
     4mm2gb   scsi 2.0 GB 4mm Tape Drive
     4mm4gb   scsi 4.0 GB 4mm Tape Drive
     525mb    scsi 525 MB 1/4-Inch Tape Drive
     8mm      scsi 2.3 GB 8mm Tape Drive
     8mm5gb   scsi 5.0 GB 8mm Tape Drive
     8mm7gb   scsi 7.0 GB 8mm Tape Drive
     9trk     scsi 1/2-inch 9-Track Tape Drive
     ost      scsi Other SCSI Tape Drive
```

This list comes from the predefined device object class. If you are adding a non-IBM SCSI tape drive, choose **`ost scsi Other SCSI Tape Drive`**. In fact, any time you add an non-IBM SCSI device of any kind, there is always a generic SCSI option in SMIT for you to use. For our example, we will choose **`8mm5gb scsi 5.0 GB 8mm Tape Drive`**. This will produce a dialog asking to choose a parent adapter:

```
                          Parent Adapter

   Move cursor to desired item and press Enter.

     scsi0 Available 00-00-0S Standard SCSI I/O Controller
```

Unless you have more than one adapter, you would choose **`scsi0 Available 00-00-0S Standard SCSI I/O Controller`**.

Finally, you will get the following screen:

```
                          Add a Tape Drive

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
  Tape Drive type                                  8mm5gb
  Tape Drive interface                             scsi
  Description                                      5.0 GB 8mm Tape Drive
  Parent adapter                                   scsi0
* CONNECTION address                              []               +
  BLOCK size (0=variable length)                  [1024]           +#
  Use DEVICE BUFFERS during writes                 yes             +
  Use EXTENDED file marks                          no              +
  DENSITY setting #1                               140             +
  DENSITY setting #2                               20              +
  Use data COMPRESSION                             yes             +
```

Here the only required entry is the **`CONNECTION address`**, which is your SCSI address. Other parameters you can change as needed. Herein is the advantage of adding a device through SMIT: you can customize your configuration. Otherwise, using **`cfgmgr`** will add the device with default attributes. If the tape drive in this example is set to SCSI ID 6, you would enter 6,0 in the **`CONNECTION address`** field.[2] Pressing the **`Enter`** key at this point will execute the **`mkdev`** command and this will configure the drive to an available state.

Adding Unsupported Devices

The predefined object class supports all kinds of devices, most of which are IBM products. However, it does contain generic device objects. To see a list of those objects, enter the following command:

```
# lsdev -P | grep -i other
printer        osp          rs232      Other serial printer
printer        osp          rs422      Other serial printer
printer        opp          parallel   Other parallel printer
tape           ost          scsi       Other SCSI Tape Drive
cdrom          oscd         scsi       Other SCSI CD-ROM Drive
cdrom          scsd         scsi       Other SCSI CD-ROM Drive
disk           osdisk       scsi       Other SCSI Disk Drive
disk           scsd         scsi       Other SCSI Disk Drive
rwoptical      osomd        scsi       Other SCSI Read/Write Optical
rwoptical      scsd         scsi       Other SCSI Read/Write Optical
```

If your device is not part of the predefined object class and closely resembles one of the above objects, then using one of them when adding a device through SMIT may very well work. Often **`cfgmgr`** will do just that when it encounters an unknown device. However, if this doesn't work you have no choice but to modify the predefined object database, write the necessary device methods, write a device driver (if necessary), and install the software to support the driver. Consult *Kernel Extensions and Device Programming Concepts* for more information.

Removing Devices

When you physically remove a device from the system or power it off and leave it off, **`cfgmgr`** resets the device state to **`defined`** the next time a reboot occurs in case you want to reattach it later. If you want to set the device state to **`defined`** without rebooting the system, use the **`rmdev`** command. The syntax of the **`rmdev`** is:

```
rmdev -l Name
```

`Name` is the logical name of the device. To find the logical names of the customized devices, use the **`lsdev -C`** command. The first column of this command's output is the logical names of the devices. Thus, to remove our example 8mm drive you type:

```
# rmdev -l rmt0
```

This keeps the **`rmt0`** device configuration information in the customized database but sets its state to **`defined`**. To remove the drive entirely from the customized database, type:

```
# rmdev -d -l rmt0
```

[2]The form of *x,y* for the address is *x* being the SCSI ID and *y* being the logical unit number (LUN).

The above can be done through SMIT. Just type **`smit device`**, select your device, and then choose the appropriate menu selection to remove it. You will be prompted as to whether or not you want to keep the device in the customized database.

HP-UX

Device Files Introduction

For a peripheral device to work in HP-UX 10, the following must be true:

- The device must be connected to the computer and turned on.
- The appropriate drivers must be part of the kernel.
- The drivers must be connected in the proper order.
- At least one device file must exist for the device.

This information also needs to be mapped in a way that allows the kernel to associate a device file with the appropriate hardware address and driver. When you configure system hardware you inform the operating system what hardware is present. Much configuration is done automatically when you boot the system. How much you actually have to do depends on whether the device is autoconfigurable and whether the driver is present in the currently running kernel. If neither is the case, use SAM or HP-UX commands to add the device.

At system boot, the kernel performs several system initialization tasks, including probing all hardware installed on the system. During the hardware probe, the kernel identifies all devices—buses, channel adapters, device adapters, and external devices—that can be autoconfigured. The kernel binds an appropriate driver to each device detected at a specific hardware address. This only happens for autoconfigurable devices.

After completing system initialization tasks, including hardware probing, the kernel invokes the **`init`** command. **`init`** reads the **`/etc/inittab`** file and invokes several system startup commands listed in the file, including **`/sbin/ioinitrc`**. This command usually starts **`ioinit`**, which does several things:

- Reads the contents of the **`/etc/ioconfig`** file and transfers device binding information found there to the kernel data structures, **`io_tree`**.
- Executes **`insf`**, which creates device files, if necessary.

All Hewlett-Packard peripheral devices supported by HP-UX 10 are automatically configurable. Device files for devices or I/O cards are automatically created during the reboot process.

By convention device files are maintained in the **`/dev`** directory. Some device files are found in **`/dev`** itself while others are grouped in its subdirectories. Device files defined in subdirectories are grouped by device type (reel tape, cartridge tape, etc.) and by device class (block or character). The following list shows some of the devices files and subdirectories in **`/dev`**:

`/dev/ac`	Block device files for magneto-optical drives
`/dev/rac`	Character device files for magneto-optical drives
`/dev/dsk`	Block device files for disk sections and LVM disks
`/dev/rdsk`	Character device files for disk sections and LVM disks
`/dev/vg`*`nn`*	Directory containing device files for logical volumes in a volume group
`/dev/vg`*`nn`*`/lvol`*`n`*	Block device files for logical volumes
`/dev/vg`*`nn`*`/rlvol`*`n`*	Character device files for logical volumes
`/dev/ct`	Clock device files for cartridge tape drives
`/dev/rct`	Character device files for cartridge tape drives
`/dev/mt`	Block device files for 1/2-inch reel and DDS tape
`/dev/rmt`	Character device files for 1/2-inch reel and DDS tape
`/dev/ptym`	Master pseudo terminal device files
`/dev/pty`	Slave pseudo terminal device files
`/dev`	All other device files, including those for terminals, modems, and printers

Device File Naming Conventions

Disks

Within the **`/dev/dsk`** and **`/dev/rdsk`** directories, the following naming convention applies:

```
/dev/[r]dsk/cCtTdD[sS]
```

where,

`dsk`	denotes the directory containing the block device files for disks
`r`	denotes the directory containing the character (raw) device files for disks
`C`	is the controller, referencing the controller on the system to which the disk drive is connected. This number will be the same for all disks connected to that controller.
`T`	is the target number. On a SCSI bus, for example, each disk has its unique target number.
`D`	is the hardware device unit number. This is only important for disk/tape products that have two or more devices with a shared controller. For products not on a shared controller, the device unit number will always be 0. For products on shared controller the numbers reference the internal number of the device units.
`S`	is the section number of the disk. By default, the insf command will not create device files for all sections of a disk. If you don't want to use the logical volume manager you must create the device files for the diefferent disk sections manually with mksf.

Examples

`/dev/dsk/c0t6d0`	Block special file for disk 6 on disk controller 0
`/dev/dsk/c0t6d0s1`	Block special file for section 1 of disk 6 on disk controller 0
`/dev/rdsk/c1t5d0`	Character special file for disk 5 on disk controller 1

LVM devices

The Logical Volume Manager allows you to partition disks in a manner similar to, but more flexible than, disk partitioning. Using logical volumes you can combine one or more disks (physical volumes) into a volume group, which is then subdivided into logical volumes.

Physical volumes

Physical volumes are identified by their device file names, for example **`/dev/dsk/c5t3d0`** and **`/dev/rdsk/c5t3d0`** for disk 3 on disk controller 5. Note that there is a block device file and a character or raw device file, the latter identified by the "**`r`**". Which name you use depends on what task you are doing with the disk.

Use a physical volume's raw device file for these two tasks only:

- When creating a physical volume.
- When restoring your volume group configuration.

For all other tasks, use the block device file. For example, when you add a physical volume to a volume group, you use the disk's block device file for the disk, such as **`/dev/dsk/c5t3d0`**.

Volume groups

By default directories for volume groups are numbered **`vg00, vg01, ... vg`*`nn`*** in the order they were created. These are regular directories, not device files.

Logical volumes

When assigned by default, these names take the form:

`/dev/vg`*`nn`*`/lvol`*`n`*	the block device file form
`/dev/vg`*`nn`*`/rlvol`*`n`*	the character device file form

where,

`nn`	is the number of the volume group
`n`	is the number of the logical volume

When LVM creates a logical volume, it creates both block and character device files. LVM then places the device files for a logical volume in the appropriate volume group directory. For example, the default block name for the first logical volume created in volume group **`vg01`** would have the full path name:

```
/dev/vg01/lvol1
```

If you create a logical volume to contain raw data for a sales database, you might want to name it using a non-default name:

```
/dev/vg01/sales_db_lv
```

After the logical volume in the above example has been created, it will have two device files:

`/dev/vg01/sales_db_lv`	the block device file
`/dev/vg01/rsales_db_lv`	the character, or raw, device file

9-Track Reel Tapes and DDS/DATs

Tape device files use the same mechanism to select the target as disk naming does. After selecting the tape drive in the name, the options for this device are named. To simplify the use of tape device file names, the **`insf`** command automatically creates more than one device file. 9-track reel tapes can be written with four densities: 800 bpi, 1600 bpi, 6250 bpi, and compressed. DDS/DAT tape drives support only two different densities: compress or noncompressed.

Tape device files take the following form:

```
/dev/[r]mt/cCtTdD[options]
/dev/[r]mt/T[options]
```

where,

C	is the tape drive controller number
T	is the target number of the tape drive
D	is the hardware device number

Options include:

`BEST`	sets the best known options for this device, including hardware compression on all devices supporiting compression
`h|m|l`	specifies the density at which the tape is to be written or read. **`l`** indicates low density (800 bpi), **`m`** indicates medium density (1600 bpi), and **`h`** indicates high density 6250 bpi).
`n`	indicates that the tape will not be rewound or repositioned in any way when the device file is closed
`c`	indicates data compression
`b`	indicates Berkeley-style access format when reading

Examples

/dev/rmt/c1t0d0BEST	Character device file for reading or writing a tape at best density drive 0 on controller 1
/dev/rmt/c1t0d0BESTnb	Character device file for reading or writing a tape at high density with no rewind on close, Berkeley style
/dev/rmt/0mn	Character special file for reading or writing a tape at medium denisty with no rewind on close.

Terminals, Modems, and Printers

These device files have no subdirectory of their own and are kept in the **/dev** directory. They have the following naming conventions:

```
/dev/ttyCpP
/dev/ttydCpP
/dev/culCpP
/dev/cuaCpP
/dev/lpC
```

where

C is the controller unit number of the MUX card.

P is the port number on the MUX card. Ports are numbered starting with 0.

Printers are usually numbered starting at 0 or 1.

Examples

/dev/tty0p3	Character device file for a terminal port on the first MUX at port 3
/dev/lp0	Character special file for the first printer (using a special printer card) added to the system

Major and Minor Numbers

Special files are identified by their name and major and minor numbers. You can look at this information when you do a long listing of the **/dev** directory. Consider the following command and its output:

```
#  ls -l  /dev/console  /dev/*dsk
crw--w--w-   1 root    sys        0 0x000000 Aug 15 16:11 /dev/console

/dev/dsk:
total 0
brw-r-----   1 bin     sys       31 0x002000 Aug 11 11:50 c0t2d0
brw-r-----   1 bin     sys       31 0x005000 Aug 11 13:54 c0t5d0
brw-r-----   1 root    sys       31 0x006000 Aug 11 11:56 c0t6d0

/dev/rdsk:
total 0
crw-r-----   1 bin     sys      188 0x002000 Aug 11 11:50 c0t2d0
crw-r-----   1 bin     sys      188 0x005000 Aug 11 13:54 c0t5d0
crw-r-----   1 root    sys      188 0x006000 Aug 11 11:50 c0t6d0
```

The first character of each line identifies the type of device file. A ***b*** denotes a block device, and a ***c*** indicates a character device. A block device transfers data a block at a time by using the system buffers. A disk drive holding a mountable file system is an example of a block device. A character device reads or writes data one character at a time. Tape drives are usually character devices. Some devices are capable of I/O in both block and character mode. Such devices require two device files: one for block and one for character mode. A hard disk is an example of a device which uses both character and block device files. For purposes of mounting the disk as a file system, you will use the block device file. For purposes of accessing the disk for backups, the character device file is used.

The major and minor numbers appear immediately before the date. The **`major number`** identifies what kernel driver is being referred to by the device file. The value chosen for the major number is based on both the device driver and on the access method (block or character). For devices needing both a character and block device file, there are different character major numbers and block major numbers. You can use the **`lsdev`** command to list the device drivers and their major numbers in the system. For example:

```
# lsdev
    Character     Block       Driver          Class
        0          -1         cn              pseudo
        1          -1         asio0           tty
        3          -1         mm              pseudo
       16          -1         ptym            ptym
       17          -1         ptys            ptys
       46          -1         netdiag1        unknown
       52          -1         lan2            lan
       56          -1         ni              unknown
       60          -1         netman          unknown
       64          64         lv              lvm
       66          -1         audio           audio
       69          -1         dev_config      pseudo
       72          -1         clone           pseudo
       73          -1         strlog          pseudo
       74          -1         sad             pseudo
      112          24         pflop           floppy
      116          -1         echo            pseudo
      119          -1         dlpi            pseudo
      122          -1         inet_cots       unknown
      122          -1         inet_cots       unknown
      156          -1         ptm             strptym
      157          -1         pts             strptys
      159          -1         ps2             ps2
      164          -1         pipedev         unknown
      168          -1         beep            graf_pseudo
      174          -1         framebuf        graf_pseudo
      188          31         sdisk           disk
      189          -1         klog            pseudo
      203          -1         sctl            ctl
      207          -1         sy              pseudo
      216          -1         CentIf          ext_bus
      227          -1         kepd            pseudo
      229          -1         ite             graf_pseudo
```

The first two columns list character and block major numbers respectively. A **`-1`** in either column means that a major number does not exist for that type of driver. Note that in the example above the SCSI disk driver (**`sdisk`**) has both a character and a block major number.

Thus a long listing of the device files for such a disk at SCSI address 6 would look like the following:

```
# ls -l  /dev/*dsk/c0t6d0
/dev/dsk:
total 0
brw-r-----   1 root    sys       31 0x006000 Aug 11 11:56 c0t6d0

/dev/rdsk:
total 0
crw-r-----   1 root    sys      188 0x006000 Aug 11 11:50 c0t6d0
```

The directory containing the block device file, with a major number of 31, is **`/dev/dsk`**, and the directory with the character device file in it is **`/dev/rdsk`**. (The letter ***`r`*** is used here because character mode is often referred to as "raw" mode; block mode is sometimes called "cooked" mode.)

The **`minor number`** is an encryption of address and configuration information. It typically defines one or both of the following: 1) the device's physical address, and 2) operational information, such as tape density or rewind options in the case of tape drives. Note that the minor number and the device file name are similar. There is a reason for that, which brings us to the next topic.

Creating Device Files

In most cases you dto not need to create device files. When HP-UX is first installed, the **`insf`** command creates devices files for all devices found by the system during its hardware probe. Then each time the system is rebooted, **`insf`** creates device files for any new devices that have been connected to the system. Hence, most devices you use will have a device file automatically created for them at boot time.

There are times when you need to create device files manually, such as:

- To restore device files accidently deleted
- To override standard naming conventions
- To create device files HP-UX cannot create

The mksf command

The **`mksf`** command is used to create a device file *if the device is already known to the system*. Options to **`/sbin/mksf`** include:

`-d`	Use the device driver specified. A list of device drivers is obtained with the lsdev command.
`-I`	Selects card instance
`-C`	The device specified belongs to this class. The class is also obtained with the lsdev command.
`-H`	Use the hardware path specified. Hardware paths are obtained with the ioscan command.
`-D`	Override the default device installation directory and install in directory named
`-r`	Create a character (raw) device file
`-v`	Use verbose output

Examples

To create character device file **`/dev/rmt/1hn`** for a 6250bpi mag tape drive with a norewind flag on card instance 1:

```
 # mksf -d tape2 -I 1 -b 6250 -n -r /dev/rmt/1hn
```

To create a device file name **`/dev/pr6001pm`** for an HP 2564B printer (operating with 600 lines per minute):

```
 # mksf -d lpr2 -I 1 /dev/pr6001pm
```

To create a device file for a dial-in terminal with CCITT protocol on port 5 of the first MUX:

```
 # mksf -d mux -I 0 -p 5 -c -I
```

To create a device file named **`/dev/printer`** and map it to the line printer with card instance 2:

```
 # mksf -C printer -I 2 /dev/printer
```

To create a device file using the default naming conventions for the tape tape device at hardware path 56/52.0.0 for raw mode, 1600 bits per inch, and no rewind of close:

```
 # mksf -H 56/52.0.0 -r -b 1600 -n
```

The insf command

The **`insf`** command is used to create a device file *if the device has not been assigned yet*. It creates the device file and also obtains a card instance number for the device. Options to **`/sbin/insf`** include:

`-d` Use the device driver specified. A list of device drivers is obtained with the **`lsdev`** command.

`-C` The device specified belongs to this class. The class is also obtained with the **`lsdev`** command.

`-H` Use the hardware path specified. Hardware paths are obtained with the **`ioscan`** command.

`-I` Selects card instance

`-k` Assign card instance numbers but do not create device file names

`-e` Creates device files for existing devices

`-f` Forces creation of device files

`-D` Override the default device installation directory and install special file in *directory*

With **`insf`** device files can be made for all devices on your system. In addition, devices files can be made for just one particular device type (driver name) or just an individual device within a device type. You cannot specify special options with **`insf`**. If you have some device that requires special options, you need to use **`mksf`** after running **`insf`**.

Examples

To install all device files needed for the devices on your system, simply type the following:

```
# insf
```

To add an additional disk with the card instance number 2, issue the following command:

```
# insf-evd disc1 -I 2
insf: Installing special files for disc1 instance 0 address 48.2
     making dsk/c0t2d0
     making rdsk/c020t0
     making diag/dsk/c020t0
     making diag/rdsk/c020t0
     making ct/c020t0
     making ct/c0t2d0
     making ct/c0t2d1
     making rct/c020t0
     making rct/c0t2d1
     making diag/rct/c0t2d0
     making diag/rct/c0t2d1
```

If you want to exented your system by adding a second multiplexer interface board at card instance 1, issue the following command:

```
 # insf -d mux0 -I 1
 insf: Installing special files for mux0 instance 1 address 44
      making diag/mux1
      making tty1p0 tty1p1 tty1p2 tty1p3 tty1p4 tty1p5
      making diag/ tty1p- tty1p1 tty1p2 tty1p3 tty1p4 tty1p5
```

To create device files for all devices of the **`tty`** class:

```
 # insf -C tty
```

To create device files for the device at address 4.2.0:

```
 # insf -H 4.2.0
```

mksf v. insf

The **`mksf`** command creates only one device file for a device. Use **`mksf`** to create a single device file that does not use standard naming conventions and to create device files that **`insf`** could not create.

The **`insf`** command automatically creates default device files for all new devices and also assigns card instance numbers at boot time. Use **`insf`** to create multiple device files at one time using standard naming conventions.

Modifying the Kernel in Order to Add a Device

For most systems, the default kernel configuration is sufficient and you will not have to modify the kernel configuration in order to add a device. However, kernel drivers such as **`cs80`** (CS80 disk driver) and **`hshpib`** (high-speed HP-IB device driver) are not included in the default configuration. These drivers are needed for the optional EISA HP-IB card and its peripherals (for example, to communicate with an HP-IB DDS-format drive). To configure them, you must rebuild the kernel.

Rebuilding the Kernel Manually

Below is the basic procedure for manually rebuilding the kernel. For more detailed information see chapter 1 of *HP-UX System Administration Tasks*.

1. Change directory to the build environment (**`/stand/build`**). There, execute a system preparation script, **`system_prep`**. **`system_prep`** writes a system file based on your current kernel in the current directory. (That is, it creates **`/stand/build/system`**.) The **`-v`** provides verbose explanation as the script executes.

   ```
   # cd /stand/build

   # /usr/lbin/sysadm/system_prep -v -s system
   ```

2. Edit the **`/stand/build/system`** file to add the absent driver(s).

3. Build the kernel by invoking the **`mk_kernel`** command. This creates /**`stand/build/vmunix_test`**, a kernel ready for testing.

   ```
   # /usr/sbin/mk_kernel -s system
   ```

4. Save the old system file and kernel by moving them. Thus, if anything goes wrong, you still have a bootable kernel.

   ```
   # mv /stand/system /stand/system.prev
   ```

   ```
   # mv /stand/vmunix /stand/vmunix.prev
   ```

5. Move the new system file and new kernel into place, ready to be used when you reboot the system.

   ```
   # mv /stand/build/system /stand/system
   ```

   ```
   # mv /stand/build/vmunix_test /stand/vmunix
   ```

6. Shut down and halt the system using the **`/usr/sbin/shutdown -h`** command.

7. When HALTED, you may cycle power appears on the screen, turn off the computer and unplug the power cord. This is recommended for all devices; for SCSI devices and interface cards, it is required.

8. Install the peripheral device, following directions in the supplied hardware documentation.

9. Power on the peripheral devices and wait for them to signal ready; then power on the computer system, which will cause your system to reboot. As HP-UX reboots, it will create the device special files required by the new peripheral device in the appropriate **`/dev`** directories.

Using SAM to Rebuild the Kernel

To use SAM to reconfigure the kernel, log in as the superuser, ensure you are logged on to the machine for which you are regenerating the kernel and:

1. Start SAM.

2. Select the **`Kernel Configuration`** menu item.

3. Select **`Drivers`**. The following information is provided:

   - Name of the driver

   - Current state of the driver: whether it is In or Out of **`/stand/vmunix`**. In means the driver is part of the kernel. Out means the driver is not part of the kernel.

   - Pending state of the driver: whether it is In or Out of the kernel to be built. In means the driver is pending to be part of the kernel. Out means it is pending to be removed from the kernel.

   - Description of the driver

4. Using the **`Actions`** menu, select one of the drivers and add or remove it.

5. Select **`Create a New Kernel`** from the **`Actions`** menu.

6. When the new kernel has been created, you will be given the choice of moving the kernel into place and rebooting the system immediately, or exit without moving the kernel into place.

Summary

Regardless of how device files are created, both AIX and HP-UX use device files in the same way. These files are special files that provide user-level access to device drivers, so commands like **`tar`** can be used to read to and write from tape drives. For example, in AIX you can archive files in the **`/home`** directory to tape by typing:

```
 # tar -cvf /dev/rmt0 /home
```

Or in HP-UX:

```
 # tar cvf /dev/rmt/0m /home
```

All you have to do is remember the locations of the device files on the respective systems. The following tables will help you do that for the most commonly used devices.

AIX File System Device Files

Block	Character
/dev/hd1	/dev/rhd1
/dev/hd2	/dev/rhd2
/dev/hd3	/dev/rhd3
/dev/hd*	/dev/hd*

HP-UX File System Device Files

Block	Character
/dev/dsk/c0t0d0	/dev/rdsk/c0t0d0
/dev/dsk/c0t1d0	/dev/rdsk/c0t1d0
/dev/dsk/c0t2d0	/dev/rdsk/c0t2d0
/dev/dsk/c0t*d0	/dev/rdsk/c0t*d0

AIX Tape Drive Device Files

File	Meaning
/dev/rmt*	Rewind on close, no retension on open, high density
/dev/rmt*.1	No rewind on close, no retension on open, high density
/dev/rmt*.2	Rewind on close, retension on open, high density
/dev/rmt*.3	No rewind on close, retension on open, high density
/dev/rmt*.4	Rewind on close, no retension on open, low density
/dev/rmt*.5	No rewind on close, no retension on open, low density
/dev/rmt*.6	Rewind on close, retension on open, low density
/dev/rmt*.7	No rewind on close, retension on open, low density

HP-UX Tape Drive Device Files

File	Meaning
/dev/rmt/*m	AT&T best density available
/dev/rmt/*mb	Berkeley best density available
/dev/rmt/*mn	AT&T no rewind best density available
/dev/rmt/*mnb	Berkely no rewind best density available
/dev/rmt/c0t*d0BEST	AT&T best density available
/dev/rmt/c0t*d0BESTb	Berkeley best density available
/dev/rmt/c0t*d0BESTn	AT&T no rewind best density available
/dev/rmt/c0t*d0BESTnb	Berkeley no rewind best density available

For AT&T-style devices the tape is positioned after the EOF following the data just read. For Berkeley-style devices, the tape is not repositioned in any way.

AIX Floppy Drive Device Files

Block	Character
/dev/fd0	/dev/rfd0

HP-UX Floppy Drive Device Files

Block	Character
/dev/floppy/c0t1d0	/dev/rfloppy/c0t1d0

5. Disks and File Systems

Directory Structures

Directories common to both AIX and HP-UX

Directory	Purpose
/dev	Used for special files for local devices.
/etc	Contains host-specific system and application configuration files important to the correct operation of the system. Most commands previously located in /etc can be found in /usr/sbin.
/export	Used to support diskless file sharing. Servers export root directory hierarchies for networked clients.
/home	User files and directories. In previous versions of AIX and HP-UX this was /u and /users, respectively.
/lost+found	Contains files located by fsck.
/mnt	Reserved name for mount points for local file systems.
/sbin	Commands and scripts essential to boot the system and mount the /usr file system. In AIX most commands used in a boot reside in memory, so /sbin contains very few commands. In HP-UX, /sbin contains quite a bit of stuff, including duplicates of commands found in /usr/bin and /usr/sbin and files needed to shut down a system.
/tmp	System-generated temporary files.
/usr	The bulk of the operating system, including commands, libraries, and documentation. The /usr file system contains only shareable operating system files, such as executables and ASCII documentation. Multiple systems of compatible architectures should be able to access the same /usr directories.
/usr/bin	Commonly used commands. In both operating systems /bin is a symbolic link to here.
/usr/ccs	For HP-UX the minimal C compiler is located here. For AIX, contains unbundled development package binaries.
/usr/include	Header files.
/usr/lbin	Backends to commands in the /usr hierarchy.
/usr/lib	Holds libraries. /lib is now a link to this directory in both operating systems.
/usr/sbin	System administration commands.
/usr/share	Architecture-independent shareable files.
/usr/share/dict	spell and ispell dictionaries. In AIX /usr/dict is a symbolic link to this directory.

/usr/share/man	man pages. In AIX /usr/man is a symbolic link to this directory.
/var	Multipurpose log, temporary, transient, variable sized, and spool files.
/var/adm	Common administrative files, logs, and databases. /usr/adm is a link to here in both operating systems.
/var/preserve	Files preserved by vi. In both operating systems /usr/preserve is a symbolic link to /var/preserve.
/var/spool	Host-specific spool files. In both operating systems /usr/spool is a link to here.
/var/tmp	User temporary files generated by commands in the /usr hierarchy. Both operating systems have symbolic links from /usr/tmp.

AIX-only Directories

Directory	**Purpose**
/tftpboot	Boot images and information for diskless clients.
/usr/lpp	Optional applications.
/var/spool/mail	Mail spool directory. /usr/mail is a symbolic link to here.
/usr/lib/lpd	Programs and configurations for AIX printing. /usr/lpd is a link to here.

HP-UX-only Directories

Directory	**Purpose**
/etc/opt	Applications will store application-specific, host-specific configuration data under /etc/opt/application.
/etc/rc.config.d	Configuration data files for startup and shutdown scripts.
/opt	Optional applications.
/sbin/init.d and /sbin/rc#.d	All rc scripts used to start up and shut down various subsystems.
/stand	System-specific kernel configuration and binary files.
/usr/conf	Shareable kernel build environment.
/usr/contrib	Contributed software.
/usr/local	Site-specific files, including binaries, libraries, sources, and documentation.
/usr/newconfig	Default operating system configuration data files.
/usr/old	Used for host customization during an operating system update. System files replaced by files in /usr/newconfig will be moved here.
/var/mail	Mail spool directory. /usr/mail is a symbolic link to this directory.

AIX File Systems

Logical Volume Manager

Before you can understand the way in which AIX handles file systems, you must have a basic understanding of the LVM, Logical Volume Manager. LVM is a disk management mechanism that represents a significant departure from traditional UNIX partitioning schemes. Among its advantages is the ability to allocate additional space to a file system without having to rebuild the disk. There are a number of components to LVM: physical volumes (PVs), volume groups (VGs), physical partitions (PPs), logical volumes (LVs), and logical partitions (LPs).

Basic facts

- A **physcial volume** is a hard disk.
- A **volume group** consists of one or more physical volumes, for a maximum of 32.
- Data is written to physical volumes in 4MB chunks (the default size) known as **physical partitions**.
- Physical partitions can be allocated to a **logical volume**, which usually contains a file system, but these physical partitions can come from different places on the disk(s).

- You can specify that your data be "striped" when creating a logical volume. Striping distributes data blocks evenly across multiple disks to improve read and write performance, but normally logical volumes do not handle data in this fashion .
- Physical partitions, which are scattered throughout the disk(s), are represented to logical volumes via sequentially numbered **`logical partitions`**. There is a one-to-one relationship between physical partitions and logical partitions, unless you are doing disk mirroring, in which case you would probably have two logical partitions for every one physical partition . When you create a logical volume, you allocate a certain number of logical partitions to it. Therefore, allocating 64 logical partitions to a logical volume creates a 256MB logical volume, since the default size of the logical partitions (and its accompanying physical partitions) is 4 MB.

Journaled File Systems

Logical volumes can and do contain paging spaces and dump areas, but most often they contain file systems. The structure of AIX file systems very much resembles those of other UNIX systems. Known as journaled file systems (JFS), AIX file systems each contain a superblock, inodes, and data blocks. Journaled file systems use a log system instead of **`fsck`** to reinstate a file system after a failure. A JFS is not confined to an entire disk, as in HP-UX's hierarchical file system (HFS), and its contents do not come from a partition of contiguous disk space. JFS writes data in 4K blocks but supports BSD-style fragmentation to improve disk utilization. JFS also supports file compression and decompression using the LZ algorithm.

AIX Disk Installation and File System Creation

The following discussion will use as an example an RS/6000 system with one internal hard disk to which we will add an external hard disk.

For the system to recognize the existence of the new disk, all you have to do is power down the system, connect the disk, and reboot. The new disk will automatically be assigned a physical volume name of **`hdisk`**<**`x`**>, where **`x`** is the next available disk number. For example, the internal disk's label would be **`hdisk0`**; the newly added external drive would be named **`hdisk1`**. At this point the disk is available to LVM but not configured. Configuring the drive can be done through SMIT or with the **`chdev`** command.

If you wanted to connect the external hard drive using SMIT, type **`smit makdsk`** at the command line. You will see the following:

```
                               Disk Type

Move cursor to desired item and press Enter.

[TOP]

  1000mb        scsi 1.0 GB SCSI Disk Drive
  1000mb16bit   scsi 1.0 GB SCSI Disk Drive
  1000mb16bitde scsi 1.0 GB 16 Bit Differential SCSI Disk Drive
  1000mbde      scsi 1.0 GB Differential SCSI Disk Drive
  1100mb        scsi 1.1 GB SCSI Disk Drive

  .......

  540mb         scsi 540 MB SCSI Disk Drive
  670mb         scsi 670 MB SCSI Disk Drive
  730mb         scsi 730 MB SCSI Disk Drive
  857mb         scsi 857 MB SCSI Disk Drive
  osdisk        scsi Other SCSI Disk Drive
  scsd          scsi Other SCSI Disk Drive
```

If you are installing a non-IBM SCSI disk, choose **`osdisk`** from the **`Disk Type`** listing, select the appropriate adapter in the **`Parent Adapter`** listing, and then enter the SCSI address in the **`CONNECTION address`** field in the **`Add a Disk`** screen:

```
                               Add a Disk

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
  Disk type                                      osdisk
  Disk interface                                 scsi
  Description                                    Other SCSI Disk Drive
  Parent adapter                                 scsi0
* CONNECTION address                            []                      +
  ASSIGN physical volume identifier              no                     +
  Queue DEPTH                                   [1]                     +
  Queuing TYPE                                  [none]                  +
  Use QERR bit                                  [yes]                   +
  Device CLEARS its Queue on error              [no]                    +
  READ/WRITE time out value                     [30]                    +
  START UNIT time out value                     [60]                    +
  REASSIGN time out value                       [120]                   +
```

Note that SCSI hardware addresses for this field are a two-digit address with the second digit a zero.[3] For example, if your SCSI disk hardware address is 4, then your connection address is 4,0. For the line that reads **`ASSIGN physical volume identifier`** choose **`yes`**. At this point all you have to do is press Enter, and the disk is designated a physical volume (PV) and assigned a physical volume identifier (PVID).

[3]The form of *x,y* for the address is *x* being the SCSI ID and *y* being the logical unit number (LUN).

The next step is to make the PV part of a volume group (VG). It is only within the context of a volume group that you can create a logical volume. If you create a volume group containing both the internal and external disks, then the two disks are considered a single disk entity by LVM.

There is a special volume group called **`rootvg`**. **`rootvg`** contains, among other things, the operating system and boot area, which in many cases should reside on only one disk. If you were to spread the **`rootvg`** over both your internal and external disks, then failure of just one of the disks would require you to reload the operating system and reconfigure it as well as restore user data from backups.

The new disk can be in its own volume group, or you can add it to an existing volume group, even **`rootvg`** if you so choose. In our example so far of a machine with two disks, one internal and one external, if you chose to add **`hdisk1`** to the **`rootvg`**, you would use SMIT (**`smit extendvg`**) or just the **`extendvg`** command:

```
# extendvg -f rootvg hdisk1
```

If you wanted to make physical volume **`hdisk1`** its own volume group called **`datavg`**, you would use SMIT (**`smit mkvg`**) or the **`mkvg`** command:

```
# mkvg -f -y datavg hdisk1
```

You must use the **`varyonvg`** command to activate the volume group before you access it:

```
# varyonvg datavg
```

If you were to put **`hdisk1`** into the **`datavg`** this way, then the **`lsvg`** (list volume groups) command, given our example so far, would produce:

```
# lsvg
rootvg
datavg
```

If you were to type the **`lspv`** (list physical volumes) command, you would get something like:

```
# lspv
hdisk0     0004038485e2483a  rootvg
hdisk1     000013403f203af5  datavg
```

The first column would be the physical volume label, the second column would be the hexadecimal PVID, and the third would be the volume group to which each physical volume belongs.

With the new disk now recognized by the system as a physical volume belonging to the **`datavg`** volume group, it is now possible to create a logical volume. Before proceeding, it is worth noting that you can customize the creation of a logical volume to a great extent. For example, you can determine how many disks an LV can span, or whether the LV should be confined to inner edges, center, or outer edges of a particular disk. However, LVM will provide default values for these items if you prefer not to specify them yourself, and for purposes of this book we will assume the defaults.

You can create a logical volume with the **`mklv`** command, but it's easier to create it using SMIT. Just type **`smit mklv`** at the command line. You will first be asked to provide the volume group name (which in our case would either be **`rootvg`** or **`datavg`**), and then once you do you will see something like the following:

```
                           Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                [Entry Fields]
  Logical volume NAME                              []
* VOLUME GROUP name                                 rootvg
* Number of LOGICAL PARTITIONS                     []                    #
  PHYSICAL VOLUME names                            []                   +
  Logical volume TYPE                              []
  POSITION on physical volume                       middle              +
  RANGE of physical volumes                         minimum             +
  MAXIMUM NUMBER of PHYSICAL VOLUMES               []                    #
    to use for allocation
  Number of COPIES of each logical                  1                   +
    partition
  Mirror Write Consistency?                         yes                 +
  Allocate each logical partition copy              yes                 +
  Mirror Write Consistency?                         yes                 +
  Allocate each logical partition copy              yes                 +
    on a SEPARATE physical volume?
  RELOCATE the logical volume during                yes                 +
    reorganization?
  Logical volume LABEL                             []
  MAXIMUM NUMBER of LOGICAL PARTITIONS             [128]
  Enable BAD BLOCK relocation?                      yes                 +
  SCHEDULING POLICY for writing logical             parallel            +
    partition copies
  Enable WRITE VERIFY?                              no                  +
  File containing ALLOCATION MAP                   []
  Stripe Size?                                     [Not Striped]        +
```

In SMIT, any entry that starts with an asterisk (*), like in **`VOLUME GROUP`** name, is a required entry. So all you would have to provide in the above example is the number of logical partitions. Assuming your logical partition size is 4MB, to create a logical volume of 512MB you would fill in 128. You also have the option of providing a name, preferably one with some logic to it, for your logical volume. If not, the system will supply one for you which may be no more descriptive than **`lv01`**. In our case, we will name our LV **`data`**.

Once the logical volume is created, you can then create a file system for it with either the **`crfs`** command or by typing **`smit crfs`**. If you use SMIT command, the first thing you will see is the following:

```
                            Add a File System

Move cursor to desired item and press Enter.

Add a Journaled File System
Add a Journaled File System on a Previously Defined Logical Volume
Add a CD-ROM File System
```

If you already have defined a logical volume, as in our example thus far, you would choose the second option, **`Add a Journaled File System on a Previously Defined Logical Volume`**. Choose the first option if you want to create a logical volume and a file system at the same time. Doing so, however, gives you less control over the attributes of the new logical volume. Choosing the second option produces:

```
   Add a Journaled File System on a Previously Defined Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* LOGICAL VOLUME name                                          +
* MOUNT POINT                                       []
  Mount AUTOMATICALLY at system restart?            no         +
  PERMISSIONS                                       read/write +
  Mount OPTIONS                                     []         +
  Start Disk Accounting?                            no         +
  Fragment Size (bytes)                             4096       +
  Number of bytes per inode                         4096       +
  Compression algorithm                             no         +
```

The required entries are **`LOGICAL VOLUME name`** and **`MOUNT POINT`**. However, it is strongly recommended that you change the third option, **`Mount AUTOMATICALLY at system restart`**, to **`yes`** if you want the file system available after each boot. The mount point of a file system is simply a directory name that designates the starting point of the file system. For example, if you created a file system on a logical volume called **`data`**, and the mount point was the directory **`/usr/local/data`**, the file system would be mounted—or made available—beginning at the **`/usr/local/data`** directory. The mount point does not have to be an empty directory, but any files that exist in a mount point directory will not be seen or available once a file system has been mounted on that directory.

Which brings us to the last step. Once a logical volume has been created and a file system for that logical volume has been created, making that file system accessible requires mounting the file system. The easiest way to mount the new file system is simply to type the **`mount`** command followed by the name of the mount point, for example:

```
#  mount /usr/local/data
```

Here the newly created file system in the data logical volume is mounted on the **`/usr/local/data`** directory. AIX knows which file system to mount onto **`/usr/local/data`** because creation of a file system results in an entry in the file **`/etc/filesystems`** like the following:

```
/usr/local/data:
        dev             = /dev/data
        vfs             = jfs
        log             = /dev/hd8
        mount           = true
        check           = true
        options         = rw
        account         = false
```

The **`mount`** command will look in **`/etc/filesystems`** for an entry called **`/usr/local/data`**. If it does, and in this case it does, then it mounts whatever the **`dev`** line says (in this case **`/dev/data`**, the name of our logical volume) onto the **`/usr/local/data`** directory.

Managing Paging Space

To create paging space you create a logical volume that has the paging attribute. To add a new paging space volume:

1. Start SMIT:

   # smit mkps

   You will see the following (assuming you have only one volume group):

```
                           VOLUME GROUP name

 Move cursor to desired item and press Enter.
   rootvg
```

   After selecting the volume group you will see something like:

```
                       Add Another Paging Space

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
  Volume group name                                    rootvg
  SIZE of paging space (in logical partitions)        []              #
  PHYSICAL VOLUME name                                                 +
  Start using this paging space NOW?                   no              +
  Use this paging space each time the system is        no              +
          RESTARTED?
```

Enter the number of logical partitions. (Default is 4.) To activate the paging space now and at each system restart, choose **yes** for each of the questions listed.

To increase the size of your paging space use the **smit chps** command. To reduce your paging space you must first remove it (**smit rmps**) and then add it (**smit mkps**) at the size you want.

AIX Disk and File System Summary

The essential steps to adding a disk and creating a file system on it are:

1. Connect the disk to the system.
2. Make the disk a physical volume.
3. Add the physical volume to a volume group or create a new volume group for it.
4. Create a logical volume with the characteristics you desire, such as mirroring or particular placement on the disk.
5. Create a file system to be contained in the logical volume. If you create the file system without first creating a logical volume, then AIX will create the latter for you.
6. Mount the file system.

You can do all of this via the command line or through SMIT. Use SMIT if you are a novice or just like to make life easier for yourself.

HP-UX

Logical Volume Manager

LVM is now available on HP-UX. Prior to the 10.0 release disks were managed differently. On the Series 700, neither disk sectioning nor logical volumes were supported. Instead, disks contained only a single section that could contain a file system and optionally a swap area and a boot area. An entire disk could also be used as a raw data area, dump device, or swap area.

With the current release of HP-UX, disks are managed identically on Series 800 and Series 700 systems, and on both LVM is recommended as the preferred mechanism for managing disks. Although the use of logical volumes is encouraged, disks on both the Series 800 and Series 700 can be managed as non-partitioned disks, or with hard partitions for those disk models that support hard partitions.

Existing non-partitioned or hard-partitioned disks can be converted to use logical volumes. Both LVM disks and non-LVM disks can exist simultaneously on your system, but a given disk must be managed entirely by either LVM or non-LVM methods. That is, you cannot combine these techniques for use with a single disk.

The disk striping capabilities of Software Disk Striping on the Series 700 are no longer supported beginning at version 10 and have been replaced by disk striping on logical volumes. Existing arrays of disks that made use of SDS are automatically converted to use logical volumes during the upgrade process.

Basic Facts

- As in AIX, a **`physical volume`** is a hard disk. To use LVM, a disk must be first initialized into a physical volume (also called an LVM disk).
- Once you have initialized one or more physical volumes, you assign them to one or more **`volume groups`**. Again, this is similar to AIX's LVM.
- A given disk can only belong to one volume group. The maximum number of volume groups that can be created is determined by the configurable parameter **`maxvgs`**.
- A volume group can contain from one to 255 physical volumes.
- Disk space from the volume group is allocated into a **`logical volume`**, a distinct unit of usable disk space. A volume group can contain up to 255 logical volumes. The disk space within a logical volume can be used for swap, dump, raw data, or you can create a file system on it.
- LVM divides each physical disk into addressable units called **`physical extents`** (**`physical partitions`** in AIX). Physical extent size is configurable at the time you form a volume group and applies to all disks in the volume group. By default, each physical extent has a size of 4 megabytes (MB). This value can be changed when you create the volume group to a value between 1MB and 256MB.
- The basic allocation unit for a logical volume is called a **`logical extent`** (**`logical parition`** in AIX). A logical extent is mapped to a physical extent; thus, if the physical extent size is 4MB, so will be the logical extent size. The size of a logical volume is determined by the number of logical extents configured.

- Except for mirrored or striped logical volumes, each logical extent is mapped to one physical extent. For mirrored logical volumes, either two or three physical extents are mapped for each logical extent depending upon whether you are using single or double mirroring. For example, if one mirror copy exists, then each logical extent maps to two physical extents, one extent for the original and one for the mirror copy.
- If a logical volume is to be used for root, primary swap, or dump, the physical extents must be contiguous. This means that the physical extents must be allocated with no gaps on a single physical volume. On non-root disks, physical extents that correspond to contiguous logical extents within a logical volume can be non-contiguous on a physical volume or reside on entirely different disks. As a result, it becomes possible for a file system created within one logical volume to reside on more than one disk.

Managing Logical Volumes Using SAM

SAM enables you to perform most, but not all, LVM management tasks. Tasks that can be performed with SAM include:

- Creating or removing volume groups
- Adding or removing disks within volume groups
- Creating, removing, or modifying logical volumes
- Increasing the size of logical volumes
- Creating or increasing the size of a file system in a logical volume
- Setting up and modifying swap and dump logical volumes
- Creating and modifying mirrored logical volumes

These tasks can also be performed with HP-UX commands.

Managing Logical Volumes Using HP-UX Commands

The following tables give you general information on the commands you will need to use to perform a given task. Refer to the *HP-UX Reference* for detailed information.

Physical Volume Management Tasks

Task	Command
Changing the characteristics of a physical volume in a volume group.	pvchange
Creating a physical volume for use in a volume group.	pvcreate
Displaying information about physical volumes in a volume group.	pvdisplay
Moving data from one physical volume to another	pvmove

Volume Group Tasks

Task	Command
Creating a volume group.	vgcreate
Removing volume group.	vgremove
Activating, deactivating, or changing the characteristics of a volume group	vgchange
Backing up volume group configuration information.	vgcfgbackup
Restoring volume group configuration from a configuration file.	vgcfgrestore
Displaying information about volume group.	vgdisplay
Exporting a volume group and its associated logical volumes.	vgexport
Importing a volume group onto the system; also adds an existing volume group back into /etc/lvmtab	vgimport
Scan all physical volumes looking for logical volumes and volume groups; allows for recovery of the LVM configuration file, /etc/lvmtab.	vgscan
Adding disk to volume group.	vgextend
Removing disk from volume group.	vgreduce

Logical Volume Management Tasks

Task	Command
Modifying a logical volume	lvchange
Displaying information about logical volumes.	lvdisplay
Increasing the size of logical volume by allocating disk space.	lvextend
Decreasing the size of a logical volume.	lvreduce
Removing the allocation of disk space for one or more logical volumes within a volume group	lvremove
Preparing a logical volume to be a root, primary swap, or dump volume	lvlnboot
Removing link that makes a logical volume a root, primary swap, or dump volume.	lvrmboot
Increasing the size of a file system up to the capacity of logical volume	extendfs
Splitting a mirrored logical volume into two logical volumes	lvsplit
Merging two logical volumes into one logical volume.	Lvmerge

Creating a Logical Volume Using HP-UX Commands

To create a logical volume:

1. Select one or more disks. **`ioscan`** shows the disks attached to the system and their device file names.

2. Initialize each disk as an LVM disk by using the **`pvcreate`** command. For example, enter

   ```
   # pvcreate /dev/rdsk/c0t0d0
   ```

   Note that using **`pvcreate`** will result in the loss of any existing data currently on the physical volume and that you use the character device file for the disk. Once a disk is initialized, it is called a physical volume.

3. Create a directory for the volume group[4]. For example:

   ```
   # mkdir /dev/vg01
   ```

4. Create a device file named **`group`** in the above directory with the **`mknod`** command. The basic syntax is:

   ```
   # mknod /dev/vgnn/group c 64 0xNN0000
   ```

   The ***c*** following the device file name specifies that **`group`** is a character device file. The **64** is the major number for the **`group`** device file; it will always be 64. The **`0xNN0000`** is the minor number for the **`group`** file in hexadecimal. Note that each particular **`NN`** must be a unique number across all volume groups.

   Example:

   ```
   # mknod /dev/vg01/group c 64 0x010000
   ```

5. Create the volume group specifying each physical volume to be included using **`vgcreate`**. For example:

   ```
   # vgcreate /dev/vg01 /dev/dsk/c0t0d0 /dev/dsk/c0t1d0
   ```

   Use the block device file to include each disk in your volume group. You can assign all the physical volumes to the volume group with one command. No physical volume can already be part of an existing volume group.

6. Once you have created a volume group, you can now create a logical volume using **`lvcreate`**:

   ```
   # lvcreate /dev/vg01/lvol1
   ```

   When LVM creates the logical volume, it creates the block and character device files and places them in the directory **`/dev/vg`*****`nn`***.

File Systems

For HP-UX 10 there are now two types of local hard disk file systems you may use. Information on each is presented in the following table:

[4] See Chapter 4 for HP-UX LVM naming conventions.

File System Type	When Should I Use It?	Additional Information
HFS (High Performance File System)	When you do not have any special file system needs.	Represents HP-UX's standard implementation of the UNIX file system
VxFS (VERITAS File System)	If you require fast file system recovery and the ability to perform a variety of administrative tasks online	HP-UX's implementation of a journaled file system (JFS)

An Introduction to VxFS

HP-UX's implementation of a journaled file system, also known as JFS, is based on VxFS, the version from VERITAS Software, Inc. Prior to the version 10 release of HP-UX, HFS was the only available locally mounted read/write file system.

As compared to HFS, VxFS allows much shorter recovery times in the event of system failure. It is also particularly useful in environments requiring high performance or deal with large volumes of data. This is because the unit of file storage, called an **`extent`**, can be multiple blocks, allowing considerably faster I/O than with HFS.[5] It also provides for minimal downtime by allowing online backup and administration. You may not want to configure VxFS, though, on a system with limited memory because VxFS memory requirements are considerably larger than that for HFS.

Basic VxFS functionality is included with the HP-UX operating system software. Additional enhancements to VxFS are available as a separately orderable product called HP OnlineJFS, product number B5117AA (Series 700) and B3928AA (Series 800).

Creating a File System

When creating either an HFS or VxFS file system, you can use SAM or a sequence of HP-UX commands. Using SAM is quicker and simpler. The following provides a checklist of sub-tasks for creating a file system which is useful primarily if you are not using SAM. If you use SAM, you do not have to explicitly perform each distinct task below; rather, proceed from SAM's **`Disks and File Systems`** area menu. SAM will perform all the necessary steps for you.

You can create a file system either within a logical volume or on a non-LVM disk. However, using a logical volume is strongly encouraged. If you decide not to use a logical volume when creating a file system, skip tasks 1 through 4 below, which deal with logical volumes only.

1. Estimate the size required for the logical volume.

2. Determine if sufficient disk space is available for the logical volume within its volume group

   Use the **`vgdisplay`** command to calculate this information. **`vgdisplay`** will output data on one or more volume groups, including the physical extent size (under PE Size (Mbytes)) and the number of available physical extents (under Free PE). By multiplying

[5]An extent within VxFS should not be confused with a physical or logical extent within LVM

these two figures together, you will get the number of megabytes available within the volume group.

3. Add a disk to a volume group if necessary. If there is not enough space within a volume group, you will need to add a disk to a volume group. To add a disk to an existing volume group, use **`pvcreate`** and **`vgextend`**. You can also add a disk by creating a new volume group with **`pvcreate`** and **`vgcreate`**.

4. Create the logical volume.

   Use **`lvcreate`** to create a logical volume of a certain size in the above volume group.

5. Create the new file system.

   Create a file system using the **`newfs`** command. Note the use of the character device file. For example:

   ```
   # newfs -F hfs /dev/vg02/rlvol1
   ```

   If you do not use the **`-F`** (for file system type) option, by default, **`newfs`** creates a file system based on the content of your **`/etc/fstab`** file. If there is no entry for the file system in **`/etc/fstab`**, then the file system type is determined from the file **`/etc/default/fs`**.

   For HFS, you can explicitly specify that **`newfs`** create a file system that allows short file names or long file names by using either the **`-S`** or **`-L`** option. When creating a VxFS file system, file names will automatically be long.

6. Once you have created a file system, you will need to mount it in order for users to access it.

Mounting file systems

You can either use SAM or HP-UX commands to mount file systems. If you use SAM, proceed from SAM's **`Disks and File Systems`** area menu. You can perform the necessary tasks as part of creating your file system, as already described. For help in mounting files using SAM, see SAM's online help. This section concentrates on using HP-UX commands to mount file systems.

To mount a local file system:

1. Choose an empty directory to serve as the mount point for the file system. Use the **`mkdir`** command to create the directory if it does not currently exist. For example, enter:

   ```
   # mkdir /joe
   ```

2. Mount the file system using the **`mount`** command. Use the block device file name that contains the file system. You will need to enter this name as an argument to the **`mount`** command. For example, enter

   ```
   # mount /dev/vg01/lvol1 /joe
   ```

NOTE: If you are not using logical volumes, you may need to enter **`ioscan -fn`** to determine the block device file name to use. If the block device file does not exist, you will need to create it using **`insf`** or **`mksf`**.

Mounting File Systems Automatically at Bootup

To automatically mount a file system at bootup, list it in the **`/etc/fstab`** file. See the **`man page`** for **`fstab`** for details.

Extending the Size of a File System Within a Logical Volume

If you use SAM to increase the size of a logical volume containing a file system, SAM automatically runs **`extendfs`** for you. When using **`lvextend`** to increase the size of the logical volume container, this does not automatically increase the size of its contents. When you first create a file system within a logical volume, the file system assumes the same size as the logical volume. If you later increase the size of the logical volume using the **`lvextend`** command, the file system within does not know that its container has been enlarged. You must explicitly tell it this using the **`extendfs`** command.

Suppose the current size of a logical volume is 1024MB (1 gigabyte). Assuming the users of the file system within this logical volume have consumed 95% of its current space and a new project is being added to their work load, the file system will need to be enlarged. To increase the size of the file system, follow these steps:

1. Unmount the file system.

   ```
   # umount /dev/vg01/lvol1
   ```

2. Increase the size of the logical volume.

   ```
   # /usr/sbin/lvextend -L 1200 /dev/vg01/lvol1
   ```

   Note that the -L 1200 represents the new logical volume size in MB, not the increment in size.

3. Increase the file system capacity to the same size as the logical volume. Notice the use of the character device file name.

   ```
   # extendfs /dev/vg01/rlvol1
   ```

4. Re-mount the file system.

   ```
   # mount /dev/vg01/lvol1 /project
   ```

5. Run **`bdf`** to confirm that the file system capacity has been increased.

Disks and File Systems: AIX vs. HP-UX

AIX uses the Logical Volume Manager to manage disk space and file systems, and beginning with version 10, so does HP-UX, though the commands have different names. HP-UX also continues to support whole disk file systems. AIX uses a file called **`/etc/filesystems`** to automate the mounting of file systems at bootup; HP-UX uses **`/etc/fstab`**. Each of these files has a different format from the other. HP-UX, as does AIX, supports journaled file systems, but HP-UX also supports non-journaled file systems.

The two operating systems are alike in many ways, though sometimes in concept rather than execution. Both systems use both character and block device files for file systems, though in AIX these are created automatically when you create your logical volume. Both systems use the **`mount`** command in similar ways; both have mount points, mounts must occur in order for file systems to be accessible, and both have mechanisms for automatic mounting. Both systems' file system data structures are much the same; each has superblocks, inodes, and data blocks. However, to check the amount of space for each file system, AIX uses the **`df`** command, which displays file system sizes in 512-byte blocks, while HP-UX uses the **`bdf`** command, which displays sizes in KB.

Finally, both AIX and HP-UX file systems are available for exporting across the network via NFS (Network File Systems) in much the same manner. Therefore, if you want to share file systems between the two platforms, it doesn't really matter what the underlying method of file system creation is because exporting and mounting across the network simply requires a file system, regardless of how it got to be a file system.

6. Managing Processes

AIX

The ps Command

AIX supports both the AT&T and the BSD form of the **ps** command. To use the BSD form, simply leave off the minus sign for the command options, for example:

```
# ps alx
```

The AT&T version of the above command is:

```
# ps -elf
```

Priorities and Nice Values

Both of the above commands provide, among other things, the priority and nice values for each process. The nice value is part of the calculation for the priority value, whose range is 0 to 127. The lower the priority value, the more frequently the process is scheduled. Higher numbers mean lower priority.

The **nice** command follows the 0 to 39, again with the larger number representing the lower priority. The **nice** command syntax takes two forms: **nice -*Increment*** and **nice -n *Increment***. The latter is easier when you have to use negative values. **nice** defaults to an increment of 10 if you do not specify one. The following command increases the priority of a command by ten.

```
# nice --10 CommandName
```

The following command decreases the priority of a command by ten:

```
# nice CommandName
```

The **renice** command also takes a **-n** option. The syntax of **renice** is:

```
# renice Priority -p PID
```

If no other options are specified, then the **-p** is not required.

Signals

Like HP-UX, AIX really has two **kill** commands: **/bin/kill** and the **kill** built-in shell command. The signals for each differ. For example:

```
# /usr/bin/kill -l
NULL
HUP
INT
QUIT
ILL
TRAP
IOT
EMT
FPE
KILL
BUS
```

```
SEGV
SYS
PIPE
ALRM
TERM
URG
STOP
TSTP
CONT
CHLD
TTIN
TTOU
IO
XCPU
XFSZ
MSG
WINCH
PWR
USR1
USR2
PROF
DANGER
VTALRM
MIGRATE
PRE
GRANT
RETRACT
SOUND SAK

# kill -l
 1) HUP       14) ALRM      27) MSG       40) bad trap  53) bad trap
 2) INT       15) TERM      28) WINCH     41) bad trap  54) bad trap
 3) QUIT      16) URG       29) PWR       42) bad trap  55) bad trap
 4) ILL       17) STOP      30) USR1      43) bad trap  56) bad trap
 5) TRAP      18) TSTP      31) USR2      44) bad trap  57) bad trap
 6) ABRT      19) CONT      32) PROF      45) bad trap  58) bad trap
 7) EMT       20) CHLD      33) DANGER    46) bad trap  59) bad trap
 8) FPE       21) TTIN      34) VTALRM    47) bad trap  60) GRANT
 9) KILL      22) TTOU      35) MIGRATE   48) bad trap  61) RETRACT
10) BUS       23) IO        36) PRE       49) bad trap  62) SOUND
11) SEGV      24) XCPU      37) bad trap  50) bad trap  63) SAK
12) SYS       25) XFSZ      38) bad trap  51) bad trap
13) PIPE      26) bad trap  39) bad trap  52) bad trap
```

AIX also has a **killall** command that any user can run to kill all of his or her processes except the sending process. The syntax is:

```
# killall -Signal
```

AIX supports the **sar**, **vmstat**, and **iostat** commands. See the HP-UX section below for descriptions of these commands.

System Resource Controller

AIX has a unique way of managing processes: the System Resource Controller (SRC). The SRC takes the form of a daemon, **srcmstr**, which is started by **init** via **/etc/inittab**. **srcmstr** manages requests to start, stop, or refresh a daemon or a group of daemons. Instead of typing the name of a daemon to start it, or instead of using the **kill** command to stop a daemon, you use an

SRC command that does it for you. In this way you don't have to remember, for example, whether to use an ampersand when starting a daemon, or what signal to use when killing one. SRC also allows you to stop and start groups of related daemons with one command.

AIX has a hierarchical organization of system processes, and this organization is configured into the ODM in the form of the **`SRCsubsys`** and **`SRCsubsvr`** object classes. Daemons at the lowest levels are **`subservers`**. On a newly loaded system the only subservers are those of the **`inetd`** subsystem: **`ftp, telnet, login, finger,`** etc. To view these subservers, use the **`odmget`** command:

```
# odmget SRCsubvr
SRCsubsvr:
        sub_type = "ftp"
        subsysname = "inetd"
        sub_code = 21

SRCsubsvr:
        sub_type = "uucp"
        subsysname = "inetd"
        sub_code = 540

SRCsubsvr:
        sub_type = "telnet"
        subsysname = "inetd"
        sub_code = 23

SRCsubsvr:
        sub_type = "shell"
        subsysname = "inetd"
        sub_code = 514

SRCsubsvr:
        sub_type = "login"
        subsysname = "inetd"
        sub_code = 513

SRCsubsvr:
        sub_type = "exec"
        subsysname = "inetd"
        sub_code = 512

SRCsubsvr:
        sub_type = "finger"
        subsysname = "inetd"
        sub_code = 79

SRCsubsvr:
        sub_type = "tftp"
        subsysname = "inetd"
        sub_code = 69

SRCsubsvr:
        sub_type = "ntalk"
        subsysname = "inetd"
        sub_code = 517

SRCsubsvr:
        sub_type = "echo"
        subsysname = "inetd"
        sub_code = 7

SRCsubsvr:
        sub_type = "discard"
        subsysname = "inetd"
        sub_code = 9

SRCsubsvr:
        sub_type = "chargen"
        subsysname = "inetd"
        sub_code = 19

SRCsubsvr:
        sub_type = "daytime"
        subsysname = "inetd"
        sub_code = 13

SRCsubsvr:
        sub_type = "time"
        subsysname = "inetd"
        sub_code = 37

SRCsubsvr:
        sub_type = "comsat"
        subsysname = "inetd"
        sub_code = 1512

SRCsubsvr:
        sub_type = "bootps"
        subsysname = "inetd"
        sub_code = 67

SRCsubsvr:
        sub_type = "systat"
        subsysname = "inetd"
        sub_code = 11

SRCsubsvr:
        sub_type = "netstat"
        subsysname = "inetd"
        sub_code = 15
```

The next level is that of **`subsystem`**. In the above command, we have the **`inetd`** subsystem listed in each of the subserver stanzas. To see a list of all subsystems, use the **`odmget SRCsubsys`** command:

```
# odmget SRCsubsys
SRCsubsys:
        subsysname = "qdaemon"
        synonym = ""
        cmdargs = ""
        path = "/usr/sbin/qdaemon"
        uid = 0
        auditid = 0
        standin = "/dev/console"
        standout = "/dev/console"
        standerr = "/dev/console"
        action = 1
        multi = 0
        contact = 2
        svrkey = 0
        svrmtype = 0
        priority = 20
        signorm = 30
        sigforce = 15
        display = 1
        waittime = 20
        grpname = "spooler"

SRCsubsys:
        subsysname = "writesrv"
        synonym = ""
        cmdargs = ""
        path = "/usr/sbin/writesrv"
        uid = 0
        auditid = 0
        standin = "/dev/console"
        standout = "/dev/console"
        standerr = "/dev/console"
        action = 1
        multi = 0
        contact = 2
        svrkey = 0
        svrmtype = 0
        priority = 20
        signorm = 30
        sigforce = 31
        display = 1
        waittime = 20
        grpname = "spooler"

SRCsubsys:
        subsysname = "lpd"
        synonym = ""
        cmdargs = ""
        path = "/usr/sbin/lpd"
        uid = 0
        auditid = 0
        standin = "/dev/console"
        standout = "/dev/console"
        standerr = "/dev/console"
        action = 1
        multi = 0
        contact = 3
        svrkey = 0
        svrmtype = 0
        priority = 20
        signorm = 0
        sigforce = 0
        display = 1
        waittime = 20
        grpname = "spooler"

SRCsubsys:
        subsysname = "inetd"
        synonym = ""
        cmdargs = ""
        path = "/usr/sbin/inetd"
        uid = 0
        auditid = 0
        standin = "/dev/console"
        standout = "/dev/console"
        standerr = "/dev/console"
        action = 2
        multi = 0
        contact = 3
        svrkey = 0
        svrmtype = 0
        priority = 20
        signorm = 0
        sigforce = 0
        display = 1
        waittime = 20
        grpname = "tcpip"
...
```

Related subsystems form a **subsystem group**, the highest level of the SRC. Subsystem groups can be ascertained from the above command by means of the **grpname** descriptor. Thus the above output shows the **lpd** subsystem being part of the **spooler** subsystem group, and **inetd** a subsystem of the **tcpip** subsystem group. An easier way to view all the subsystems and subsystem groups is to use the **lssrc -a** command:

```
# lssrc -a
Subsystem         Group            PID     Status
 syslogd          ras              5456    active
 sendmail         mail             5722    active
 portmap          portmap          5994    active
 inetd            tcpip            6258    active
 biod             nfs              5276    active
 rpc.statd        nfs              8378    active
 rpc.lockd        nfs              8652    active
 qdaemon          spooler          8932    active
 writesrv         spooler          9198    active
 infod            infod            3610    active
 lpd              spooler                  inoperative
 gated            tcpip                    inoperative
 named            tcpip                    inoperative
 routed           tcpip                    inoperative
 rwhod            tcpip                    inoperative
 iptrace          tcpip                    inoperative
 timed            tcpip                    inoperative
 snmpd            tcpip                    inoperative
 dhcpcd           tcpip                    inoperative
 dhcpsd           tcpip                    inoperative
 dhcprd           tcpip                    inoperative
 nfsd             nfs                      inoperative
 rpc.mountd       nfs                      inoperative
 dtsrc                                     inoperative
 keyserv          keyserv                  inoperative
 ypbind           yp                       inoperative
 ypserv           yp                       inoperative
 ypupdated        yp                       inoperative
 yppasswdd        yp                       inoperative
```

You can also list subsystems and subsystem groups on remote hosts by using the following command:

```
# lssrc -h remote_host -a
```

The most commonly used SRC commands are **`startsrc`**, **`stopsrc`**, and **`refresh`**, each of which takes the following options:

`-s` Apply this command to a subsystem, using the subsystem name provided in the **`lssrc -a`** command

`-g` Apply this command to a subsystem group, using the subsystem group name provided in the **`lssrc -a`** command

The names of these commands imply their purpose: to start a subserver, subsystem, or subsystem group, use the **`startsrc`** command. For example, to start the **`rpc.mountd`** subsystem (which is actually the **`rpc.mountd`** daemon) type:

```
# startsrc -s rpc.mountd
```

To start the **`nfs`** subsystem group:

```
# startsrc -g nfs
```

This command starts all the subsystems (daemons) that comprise the **`nfs`** subsystem group: **`nfsd`**, **`biod`**, **`rpc.mountd`**, **`rpc.lockd`**, and **`rpc.statd`**.

To stop a subsystem or subsystem group, use the **`stopsrc`** command in exactly the same way. To stop and restart daemons, or to have daemons reread a configuration file such as **`/etc/inetd.conf`**, use the **`refresh`** command. For example:

```
# refresh -s inetd
```

Beginning with AIX version 4 the InetServ ODM object class no longer exists. Therefore making changes to **`/etc/inetd.conf`** and **`/etc/services`** no longer requires the **`inetimp`** command. In fact, the **`inetimp`**, **`inetexp`**, **`inetserv`**, and **`notinet`** commands no longer exist.

Cron

AIX supports an AT&T-style **`crontab`** file with each one-line entry containing the following:

- The minute (0 through 59)
- The hour (0 through 23)
- The day of the month (1 through 31)
- The month of the year (1 through 12)
- The day of the week (0 through 6 for Sunday through Saturday)
- The shell command

AIX also supports a convenient option to the **`crontab`** command: the **`-e`** option. This option will load the contents of your **`crontab`** file into an editing session. The editor used is determined by the value of the `EDITOR` variable. Once you save and exit from the editing session, your changes become your new **`crontab`** file and take effect immediately.

Officially, the **`crontab`** spool directories are found in **`/var/spool/cron`**, although there is a link from **`/usr/spool`** to **`/var/spool`** in AIX for compatibility with previous versions of the operating system.

HP-UX

ps, nice, and renice

The HP-UX **`ps`** command is strictly AT&T-style. Therefore a complete listing of every process is:

```
# ps -elf
```

Within a given range of priority numbers, the process assigned the lowest number has the highest priority. For example, a process assigned a priority of 1 takes precedence over a process assigned a priority of 6.

You can see at what priority a process is set to run at by looking in the `PRI` column when you invoke **`ps`** or **`top`**. (The lowest number within the range represents the highest priority.)

The following lists the four categories of priority, from highest to lowest:

1. POSIX standard priority (tunable parameter). POSIX standard priorities, known as `RTSCHED` priorities, are the highest priorities. `RTSCHED` processes have a range of priorities separate from other HP-UX priorities. The number of `RTSCHED` priorities is a user tunable parameter (**`rtsched_numpri`**), set between 32 and 512 (default 32).

2. Real-time priority (0-127). Reserved for SCHED_RTPRIO processes started with **`rtprio()`** system calls.

3. System priority (128-177). Used by system processes.

4. User priority (178-255). Assigned to user processes.

The kernel can alter the priority of time-share priorities (128-255) but not real-time priorities (0-127).

HP-UX uses the AT&T version of **`nice`** values, which run from 0 to 39 with a default of 20. 39 is the lowest priority, 0 the highest.

Note that both Korn and C shells handle **`nice`** slightly differently: **`ksh`** automatically lowers priority of background processes by four; this behavior can be modified using the **`bgnice`** argument. If you specify **`nice`** from **`ksh`**, it executes **`/usr/bin/nice`** and lowers priority by ten. If you specify nice from **`csh`**, it executes its built-in command and lowers priority by four; however, if you specify **`/usr/bin/nice`**, **`csh`** lowers priority by ten.

The **`renice`** command (**`/usr/sbin/renice`**) allows you to alter the priority of running processes. Running processes can also be altered from the Process Management area of SAM. The HP-UX version of **`renice`** has the following syntax:

```
# renice -n priority_change PID
```

The new system nice value is equal to 20 + *priority_change*, and is limited to the range from 0 through 39. If *priority_change* is a negative value, priority is increased provided the user has appropriate privileges.

HP-UX supports job control for both the POSIX, Korn, and C shells. Job control provides users with greater flexibility in managing and controlling jobs. For example, you can:

- Temporarily stop (suspend) a foreground job, by pressing CTRL-Z. This can be customized using the **`stty`** command.
- Bring a background job into the foreground, using the **`fg`** built-in shell command.
- Move a foreground job into the background, using the **`bg`** built-in shell command.

Signals

HP-UX signals look like the following. For the HP-UX **`kill`** command:

```
# /usr/bin/kill -l
NULL HUP INT QUIT ILL TRAP ABRT EMT FPE KILL BUS SEGV SYS PIPE ALRM
TERM USR1 USR2 CHLD PWR VTALRM PROF POLL WINCH STOP TSTP CONT TTIN
TTOU URG LOST CPU FSZ
```

For the shell built-in command:

```
# kill -l
 1) HUP
 2) INT
 3) QUIT
 4) ILL
 5) TRAP
 6) ABRT
 7) EMT
 8) FPE
 9) KILL
10) BUS
11) SEGV
12) SYS
13) PIPE
14) ALRM
15) TERM
16) USR1
17) USR2
18) CHLD
19) PWR
20) VTALRM
21) PROF
22) IO
23) WINCH
24) STOP
25) TSTP
26) CONT
27) TTIN
28) TTOU
29) URG
30) LOST
31) The specified trap syntax is not correct.
32) The specified trap syntax is not correct.
33) XCPU
34) XFSZ
```

The HP-UX **`killall`** command is a procedure used by **`/etc/shutdown`** to kill all active processes not directly related to the shutdown procedure. It is *not* used in the same way as in AIX. **`killall`** is chiefly used to terminate all processes with open files so that the mounted file systems are no longer busy and can be unmounted.

Starting and Stopping Subsystems

The new system startup and shutdown paradigm introduced with HP-UX 10 allows command-line control of subsystems. A subsystem in this sense is similar to that found in AIX, in which groups of related processes and procedures are known as subsystems. Subsystems can be controlled via the startup scripts found in **`/sbin/init.d`**. For example, the **`/sbin/init.d/nfs.client`** script controls daemons like **`biod`**s and issues the appropriate **`mount`** or **`umount`** commands. To start this subsystem on the command line type:

```
# /sbin/init.d/nfs.client start
```

and to stop:

```
# /sbin/init.d/nfs.client stop
```

What happens at this point also depends on what values are configured in the configuration script **`/etc/rc.config.d/nfsconf`**. This is true of all subsystems: each has a script in **`/sbin/init.d`** that can be used to start or stop the subsystem, and each has a configuration script in **`/etc/rc.config.d`** that governs precisely what happens when the former executes. For more information, see Chapter 2. Just remember that once a subsystem has been configured, it can easily be stopped or started on the command line with the following syntax:

```
# /sbin/init.d/subsystem_name start
```

or

```
# /sbin/init.d/subsystem_name stop
```

Process Management Commands

The following tools may provide additional information to help you examine your system's operation. Although they are commands, some (**`top`** and **`sar`**) are accessible from the Process Management area of SAM.

top

Displays and updates information about the top processes on the system, summarizes the general state of the system (load average), quantifies amount of memory in use and free, and reports on individual processes active on the system. Whereas **`ps`** gives a single "snapshot" of the system, **`top`** samples the system and updates its display at intervals. On multi-processing systems, **`top`** reports on the state of each CPU. This command is not available in AIX.

sar

Reports on cumulative system activity, including CPU utilization, buffer activity, transfer of data to and from devices, terminal activity, number of specific system calls used, amount of swapping and switching activity, queue lengths, and other kernel tables.

vmstat

Quantifies the use of virtual memory by processes on the system; also reports on traps and CPU activity.

iostat

Reports I/O statistics for active disks, terminal, and processor.

Cron

`crontab` files are found in **`/var/spool/cron/crontabs`** and should not be edited directly. Like AIX, there is a **`-e`** option to **`crontab`**.

The one-line entries in **`crontab`** files are the same for AIX.

Summary

AIX supports both the BSD and AT&T versions of the **`ps`** command; HP-UX does not. AIX also supports the **`nice`** and **`renice`** commands. Both operating systems have two versions of the **`kill`** command: **`/bin/ksh`** and the **`ksh`** built-in, each of which have slightly different signal values. Both systems have similar **`cron`** tables.

The biggest difference to be found with regard to process management between AIX and HP-UX is AIX's System Resource Controller, a daemon that can start, stop, or refresh a daemon or a group of daemons by means of a special set of commands. This mechanism allows for logical startup of several daemons at once as well as for orderly shutdown of daemons. Therefore, in the various "rc" files of AIX, you will see **`startsrc`** commands rather than commands to start individual daemons.

Process management between the two systems, with the exception of the System Resource Controller, is similar. To kill a process you can use the most common signals, such as SIGHUP, SIGABRT, SIGKILL, and SIGTERM, which have the same values. The shell built-in command **`kill`** and the **`bgnice`** option to the **`sh set`** command work the same way.

7. Backups

Note: The AIX examples in this chapter use a 5GB 8-mm tape drive using the **/dev/rmt0** device file, which has the following characteristics: rewind on close, no retension on open, high density. The HP-UX examples use a 4-mm DAT drive with the **/dev/rmt/0m** device file, which has rewind on close and medium density characteristics.

AIX

The backup Command

backup and **restore** are proprietary AIX commands that provide file system dumps and recovery as well as backup of individual files and directories. These commands resemble **dump** and **restore** and are provided in addition to the standard UNIX utilities of **tar**, **cpio**, **dd**, and **pax**. Doing a file system dump with the **backup** command is known as "backing up by i-node format" in AIX parlance. The basic syntax of backing up a file system by i-node is:

```
# backup -f Device -DumpLevel -u File system
```

Device can be a file or a device file. *DumpLevel* is a numeric value from 0 to 9, where 0 represents a full file system backup. The other dump levels represent incremental backups: an *n* level backup includes all files modified since the last *n* - 1 level backup. If you do not specify a dump level, **backup** defaults to a level 9 backup. The **-u** option updates the **/etc/dumpdates** file, a log of your backups used by **backup** when doing an incremental backup. You should unmount a file system before backing it up and run **fsck**, the root file system being the exception, of course. If you dump the root file system, file systems mounted on root, including journaled file systems, are *not* backed up.[6]

Examples

Do a full backup of the **/home** file system to tape, rewinding upon close:

```
# umount /home
# fsck /home
# backup -f /dev/rmt0 -0 -u /home
# mount /home
```

Back up the root file system's files that have changed since the last 0-level backup to tape, rewinding upon close:

```
# backup -f /dev/rmt0 -1 -u /
```

In the first example we used the mount point of **/home** as the name of the file system. You can do that; in fact, it is easier to remember to do it this way. However, the **/etc/dumpdates** file will record the name of the file system using the raw device file instead of the mount point. In the case of **/home** that raw device file is **/dev/rhd1** (by default the **/home** file system's device files are **/dev/hd1** and **/dev/rhd1**). Don't be confused by this. If you prefer you can use either the raw device or block device file name in the **backup** command instead of the mount

[6]See Chapter 3 for a description of journaled file systems.

point. To get the block device file name of a file system, use the **lsfs** command. From that you can determine the raw device file name by prepending the letter *r* to the block device file name.

You can use SMIT to back up of file systems by using the **smit backfilesys** fastpath command. This will enable you to back up using the **backup** command only, not **tar** or **cpio**.

To back up individual files and directories use the **-i** option (for individual files). This is also known as backing up by name. The basic syntax is:

```
# backup -f Device -i
```

If you do not include the **-f *Device*** part of this command, **backup** automatically writes to **/dev/rfd0**, the diskette drive. To use this option you must provide **backup** with a list of names as standard input by typing the names of each file after typing the command (ending the list with a Ctrl-D), using redirection, or using either the **find** command or the **cat** command. In the examples that follow the **-q** option is used to prevent AIX from prompting you to mount the backup media:

```
# backup -f /dev/rmt0 -iq
/.profile
/etc/profile
/home
^D
```

```
# backup -f /dev/rmt0 < /tmp/filelist -iq
```

```
# cat /tmp/filelist | backup -f /dev/rmt0 -iq
```

```
# find /home -print | backup -f /dev/rmt0 -iq
```

The last example above will back up the contents of the entire **/home** directory, but in the two examples before that, if you have **/home** listed in **/tmp/filelist**, only the directory name will be backed up. To construct the command in Korn shell syntax so that it backs up directory trees as well as files, type the following:

```
# find $(< /tmp/filelist) -print | backup -f /dev/rmt0 -iq
```

To see a list of the files and directories scroll on your screen while **backup** is running, use the **-v** option. To save this output, redirect standard output or use the **tee** command.

```
# find $(< /tmp/filelist) -print | backup -f /dev/rmt0 -iqv >\
/var/adm/backed.up.files
```

```
# find $(< /tmp/filelist) -print | backup -f /dev/rmt0 -iqv | tee |\
/var/adm/backed.up.files
```

You can use SMIT to back up of files by name by using the **smit backfile** fastpath command.

The restore Command

The **restore** command is used to restore files backed up by i-node or by name. This command works only on archives created by the **backup** command and is capable of determining which format, i-node or by-name, was used for the backup. The basic syntax of restoring entire file systems is:

```
# restore -f Device -r
```

The basic syntax of restoring files backed up by name is:

```
# restore -f Device -x
```

Examples

Restore an entire file system backed up by i-node from tape.:

```
# restore -f /dev/rmt0 -r
```

You do not have to unmount the file system before restoring in this manner, but be sure to change directories to the appropriate file system first because **`restore`** assumes relative path names.

To restore individual files backed up by i-node, use the **`-i`** option. This will start **`restore`** in interactive mode, which will step you through the process of restoring your files. For example:

```
# restore -f /dev/rmt0 -i
```

Extract all files backed up by name from tape:

```
# restore -f /dev/rmt0 -x
```

Extract all files from tape and display file names while doing so (**`-v`**, for verbose):

```
# restore -f /dev/rmt0 -xv
```

Extract verbosely the file **`/home/partlist`**:

```
# restore -f /dev/rmt0 -xv /home/partlist
```

Extracts the entire **`/home`** directory from tape verbosely:

```
# restore -f /dev/rmt0 -xv -d /home
```

To avoid frustration in restoring files, it is extremely important that you know whether files backed up by name were done so using full or relative pathnames. To find out, use the **`-T`** option to get a list of files, for example:

List files from the file **`/archive/backup1`** created by the **`backup`** command:

```
# restore -f /archive/backup1 -T
```

List files from tape:

```
# restore -f /dev/rmt0 -T
```

Use the names of the files or directories *exactly* as listed in the output, even if the path names begin with a "dot", such as **`./home/partlist.`**

You can use SMIT to restore files. Use the **`smit restfile`** fastpath to restore individual files and **`smit restfilesys`** to restore file systems.

Image Backups

AIX provides a means of creating a bootable tape containing an image of the root volume group. This can be a life saver for single-disk systems in that you have a very quick means of recovery in case of disk failure, provided, of course, your **`mksysb`** image is fairly current. You can also use **`mksysb`** to install other machines, although this will require redoing some of the configuration, such as IP addresses, because virtually everything on the original machine is copied to tape.

To do a system image backup:

1. Start SMIT:

```
# smit mksysb
```

2. You will see the following:

```
                           Back Up the System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
    WARNING:  Execution of the mksysb command will
              result in the loss of all material
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                              []                 +/
  Create MAP files?                                   no                +
  EXCLUDE files?                                      no                +
  Make BOOTABLE backup?                               yes               +
     (Applies only to tape)
  EXPAND /tmp if needed?                              no                +
     (Applies only to bootable tape)
  Number of BLOCKS to write in a single output    []                 #
     (Leave blank to use a system default)
```

Simply insert the device file name of your tape drive, such as **`/dev/rmt0`**, and choose **`yes`** for **`FORCE increase of work space if needed`**. The latter is necessary if your system takes more than one tape for the backup.

`mksysb` backs up only the root volume group (**`rootvg`**). Beginning with AIX version 4 you can make image backups of other volume groups by using the **`savevg`** command or using SMIT (**`smit savevg`**). For example, to back up a volume group called **`datavg`** you would enter:

```
# savevg -i datavg
```

The default device is **`/dev/rmt0`**; other devices can be specified with the **`-f`** option.

To restore a volume group you use the **`restvg`** command or the **`smit restvg`** fastpath. For example, to restore a volume group from **`/dev/rmt0`** without prompts, type:

```
# restvg -q
```

Tape Drives

It is important to understand how AIX handles tape drives because you may find yourself unable to restore files from a tape you thought contained a good backup. One of the attributes in AIX's configuration of a tape drive is that of block size, the amount of data read or written in a single operation. When a tape drive is added to AIX, a default block size is configured for you. Frequently that block size is 512 bytes. However, you can change the block size through SMIT or with the **`chdev`** command. If you have a backup tape written with a block size of 512 bytes, but your AIX driver reads in 1024-byte blocks, you will not be able to restore the contents of the tape unless you change the block size to 512 bytes. For example:

```
# chdev -l rmt0 -a block_size=512
```

If you want, you can let the application or command, such as **dd**, determine the block size, in which case you can specify that your tape drive block size is variable in length. To do so use 0 as the block size:

```
# chdev -l rmt0 -a block_size=0
```

If you are reasonably confident that you have a valid backup on tape, and you just can't read it no matter what you do, try changing the block size of the tape device until it works. Just remember: restoration of data from a tape can only work if the block size used in the restore is the same as that of backup block size.

For tape operations AIX uses the **tctl** command, similar to HP-UX's **mt** command. The syntax of **tctl** is:

```
# tctl -f Device Subcommand Count
```

The most common **tctl** subcommands are:

fsf	Moves the tape forward the number of file marks specified by Count.
bsf	Moves the tape backwards the number of file marks specified by Count.
rewind	Rewinds the tape.
erase	Erases all contents on the tape and rewinds it.
retension	Fast-forwards, then rewinds the tape

HP-UX

fbackup

Like AIX, HP-UX has a proprietary scheme for backups in addition to the standard UNIX utilities. This scheme includes the **fbackup** and **frecover** commands. The basic syntax of the **fbackup** command is:

```
# fbackup -f Device [-0-9] [-u] [-i path] [-e path] [-g graph]
```

Because **fbackup** does not by default write to standard output, the **-f** option is not optional. *Device* can be a file, device file, or a remote device file. A remote device file takes the form *machine:/dev/device_name*. You can specify **-** as the device to have **fbackup** write to standard output. The **[0-9]** option provides for incremental backups using the same scheme as that for AIX's **backup**: an *n* level backup includes all files modified since the last *n* - 1 level backup.

The **-i**, **-e**, and **-g** options provide a means of specifying which portions of a file system you want backed up. If you recall, AIX's **backup** command backs up either entire file systems, or individual files. **fbackup** allows you to include a file system with the **-i** option and exclude portions of it with the **-e** option. For example:

```
# fbackup -f /dev/rmt/Om -0 -i /usr -e /usr/tmp
```

This command does a full backup to tape of the **/usr** file system with the exception of the **/usr/tmp** directory tree. You can also place your inclusion and exclusion parameters in a graph file. A graph file is a text file containing the path names of the files and directories you either want included or excluded from your backup. These path names are preceded with either an *i* (denoting inclusion) or an *e* (denoting exclusion). For example, if your graph file contains the following lines:

```
 i /usr
 e /usr/tmp
 e /usr/lib
 i /home
 e /home/guest
```

Then the **`/usr`** file system will be backed up with the exception of the **`/usr/tmp`** and **`/usr/lib`** directory trees, and the **`/home`** file system will be backed up with the exception of the **`/home/guest`** directory tree. If the name of this graph file is **`/usr/local/backup/graph1`**, then your command would look something like:

```
 # fbackup -f /dev/rmt/0m -0 -g /usr/local/backup/graph1
```

`fbackup` uses the **`/var/adm/fbackupfiles/dates`** file to record backup information for purposes of doing incremental backups. (This is similar to AIX's **`/etc/dumpdates`** file.) This file is not updated unless you use the **`-u`** option, which is only available with the **`-g`** option.

Examples

Perform a level-8 backup to **`/dev/rmt/0m`** using the graph file **`/usr/local/graph`**, and update the **`/var/adm/fbackupfiles/dates`** file:

```
 # fbackup -f /dev/rmt/0m -8 -u -g /usr/local/graph
```

Back up everything under the current directory except **`subdir`** to **`/dev/rmt/0m`**:

```
 # fbackup -f /dev/rmt/0m -i . -e ./subdir
```

Back up everything under the current directory except **`subdir`** to **`/dev/rmt/0m`** on system **`roberts`**:

```
 # fbackup -f roberts:/dev/rmt/0m -i . -e ./subdir
```

frecover

The **`frecover`** command is the counterpart to the **`fbackup`** command. **`frecover`** can restore only files backed up with **`fbackup`**. There are four basic modes of operation for **`frecover`**.

Recover everything on a backup volume:

```
 # frecover -r
```

Extract certain files from a backup volume:

```
 # frecover -x
```

Read the index from the backup volume and write it to path:

```
 # frecover -I path
```

Restart an interrupted recovery:

```
 # frecover -R path
```

As you can see, **`fbackup`** resembles AIX's **`restore`** command's options in the form of **`-r`** and **`-x. frecover -r`** and **`frecover -x`** have some options in common:

```
frecover -x | -r [ -fhoFX ]
```

`frecover` defaults to **`/dev/rmt/0mn`**, but you can use **`-f`** to specify a different device, including a remote device. The **`-h`** option is used to restore only directories but not the files contained in them. The **`-o`** option is used to force **`frecover`** to overwrite a newer file with an older one. Normally **`frecover`** does not do this. The **`-F`** option causes **`frecover`** to strip all the leading directories from the path names of files being recovered. This allows you to restore files backed up with full path names to different directories. The **`-x`** option makes all recovered files relative to the current directory. If, for example, you restored the file **`/usr/bin/vi`** and your current directory was **`/home/root`**, then the restored file's new path would be **`/home/root/usr/bin/vi`**.

For the **`frecover -x`** mode the **`-g`** option is available. This allows you to use a graph file in the same way as with **`fbackup`**. The file format of the graph file is the same. This is useful for partial recoveries.

Examples

Recover all files from medium-density tape:

```
# frecover -rf /dev/rmt/0m
```

Recover all files indicated in **`/usr/local/graph`**:

```
# frecover -x -g /usr/local/graph -f /dev/rmt/0m
```

Retrieve an index of files from tape and put it in **`/tmp/index`**:

```
# frecover -I /tmp/index -f /dev/rmt/0m
```

HP-UX and Cartridge Tapes: tcio

The wear and tear on a cartridge tape drive is extensive when redirection is used because the data transfer rates between the host computer and the cartridge tape drive are not synchronized. Thus HP developed the **`tcio`** command to "buffer up" data transfer between the backup command and the cartridge tape drive. Instead of redirecting the output of the backup command straight to the device, it is piped through **`tcio`** to enable streaming to occur. **`tcio`** can be used with **`cpio`** as well as with **`fbackup`**.

Examples

Make relative backup on cartridge tape **`/dev/rct/0s0:`**

```
# find . -print | cpio -o | tcio -o /dev/rct/0s0
```

Back up **`/home`** to cartridge tape:

```
# fbackup -f - -i /home | tcio -o /dev/rct/0s0
```

Creating a Bootable System Backup

HP-UX doesn't have the **`mksysb`** command nor a command that all by itself does the same thing, but you can create a bootable image of a disk using DDS. The following is an example of backing up an internal SCSI disk drive at target number 6.

1. Shut down to single user mode to minimize system activity:

   ```
   # shutdown 0
   ```

2. Clear any remaining data in the buffers:

   ```
   # sync;sync
   ```

3. Use **dd** to copy the LIF boot area:

   ```
   dd if=/usr/lib/uxbootlf of=/dev/rmt/0mn bs=2k
   ```

4. Again, sync the disk:

   ```
   # sync;sync
   ```

5. Use **dd** to append to append the disk data:

   ```
   # dd if=/dev/rdsk/c0t6d0 of=/dev/rmt/0m bs=64
   ```

The first **dd** puts a boot area on the tape, making it a bootable image. Once the boot image is on tape, the tape is not rewound. The next **dd** appends an image of the SCSI disk at address 6 to the tape. Be sure to use the appropriate disk device file.

Once created, the tape can be used to completely restore the disk:

1. Insert the tape into the DDS tape drive.
2. Go into the Boot Administration utility and boot to ISL from the tape. If your tape drive is at target 3, then your command would look like:

   ```
   BOOT_ADMIN> b scsi.3.0 isl
   ```

3. Enter the following in response to the ISL prompt in order to restore to the internal drive at target 6:

   ```
   ISL> hpux restore disc(scsi.6;0)
   ```

This command restores the disk image from the tape to the actual disk at **scsi.6**, destroying any existing data on the disk. There is a 2GB limit on the amount of data that can be restored. The tape and disk must be on the boot device interface.

Interoperability Issues

Standard UNIX Utilities

Both AIX and HP-UX support the following standard UNIX utilities, each of whose behavior is nearly identical: **dump**, **rdump**, **tar**, **cpio**, **dd**, and **pax**. For doing interplatform backups it is advisable to use one of these commands. HP-UX has the standard utility **mt**; AIX does not but has the rough equivalent in **tctl**. AIX does not have the **tcio** command; instead you have to use **dd** to buffer data when using redirection. The following table summarizes the backup commands for both operating systems.

Backup Utilities Comparison

Command	**AIX**	**HP-UX**
backup	yes	no
restore	yes	no[7]
fbackup	no	yes
frecover	no	yes
dump	yes	yes
rdump	yes	yes
rrestore	yes	yes
tar	yes	yes
cpio	yes	yes
dd	yes	yes
tcio	no	yes

Interplatform Backups

The following backup commands have been tested on an HP Model 712 and an IBM RS/6000 42T.

Using tar to Copy Files from One System to the Other

You can **`tar`** a directory tree or a set of files on the HP, **`ftp`** the **`tar`** file to the RS/6000, and then unpack the file. For example,

Back up the **`.dt`** directory in the current directory to the **`/tmp/dt.tar`** archive file:

```
 # tar -cvf /tmp/dt.tar .dt
```

Copy the file to the RS/6000:

```
 # ftp RS_node
   ftp> put /tmp/dt.tar
```

Login to the RS/6000:

```
 # remsh RS_node
```

Restore the files from the **`/tmp/dt.tar`** archive:

```
 # tar -xvf /tmp/dt.tar
```

Of course you can use **`rcp`** instead of **`ftp`** or **`rlogin`** or **`telnet`** instead of **`remsh`**.

[7]This is not quite true. There is a **restore** command in HP-UX, but it is the counterpart to the **dump** command.

A quicker way to do the above is to use **remsh** and **tar** in a pipe, but this works only if your **.rhosts** is set up properly. For example:

```
# tar -cvf - .dt | remsh RS_node "(cd /tmp; tar -xvf -)"
```

This command writes the archive of **.dt** to standard output (the **-** option) and sends that output to the **tar** command on the RS/6000 and puts the unpacked files in **/tmp**.

In AIX 4 **remsh** is a hard link to **rsh** command. So the above command going the other direction would look something like:

```
# cd /usr/local/doc
# tar -cvf - doc | remsh HP_node "(cd /usr/local; tar -xvf -)"
```

This backs up the subdirectory **doc** and puts it in the HP's **/usr/local** directory.

You can also use **cpio** to copy a directory tree:

```
# find /home/guest -print | cpio -ov | remsh RS_node cpio -idv
```

This copies the HP's **/home/guest** directory to the RS/6000's **/home/guest** directory.

Using Another System's Tape Drive

For the following examples, the HP Model 712 had a 4mm DAT drive, the RS an 5GB 8mm tape drive. Since these involve using the network and the **dd** command, make sure the RS/6000 tape drive is set to variable-length blocks.

Examples using tar

Archive the HP's **.dt** directory onto the 8mm tape drive of the RS/6000:

```
# tar cvf - /.vue | remsh RS_node dd of=/dev/rmt0 obs=20b
```

Archive the RS/6000's **/doc** directory onto the 4mm DAT drive of the HP:

```
# tar cvf - /home | remsh HP_node dd of=/dev/rmt/0m obs=20b
```

Restore the contents of the tape from an RS/6000 drive to the HP:

```
# remsh RS_node dd if=/dev/rmt0 bs=20b | tar xvfb - 20
```

Restore the contents of the tape from an HP drive to the RS/6000:

```
# remsh HP_node dd if=/dev/rmt/0m bs=20b | tar -xvBf -
```

Examples using backup and restore

These examples are for using the RS/6000's backup command to write to an HP drive. HP-UX cannot restore the files; you have to use AIX.

```
# find /usr/local/data -print | backup -f - -iv | remsh HP_node dd\
of=/dev/rmt/0m

# remsh HP_node dd if=/dev/rmt/0m | restore -xvf -
```

Examples using pax

pax, short for portable archive exchange, is great for interplatform backups and restores. Both HP-UX and AIX support the **pax** command, which looks and acts the same on both systems. By default **pax** writes to standard output (with the **-w** option) and reads from standard input (with the **-r** option). Without the **-w** or **-r** option **pax** simply lists the files contained in an archive.

To archive verbosely the **/home** directory from the HP to the RS/6000's tape drive:

```
# pax -wv /home | remsh RS_node dd of=/dev/rmt0 obs=64k
```

To read the above archive tape from the RS/6000:

```
# dd if=/dev/rmt0 ibs=64k | pax
```

To restore the contents of the above archive on the RS/6000 from the RS/6000:

```
# dd if=/dev/rmt0 ibs=64k | pax -r
```

To copy the directory **/usr/local/doc** from the RS/6000 to the HP using the network:

```
# pax -wv /usr/local/doc | remsh HP_node pax -rv
```

General Tips

Variable block mode depends on the block size specified by the command itself (for example, the **-b** option in **tar**). Because other systems, including HP, often break up larger reads and writes into approximately 64K chunks, it is a good idea to keep your block sizes at no more than that amount.

Remember that if you are using an IBM 8mm tape drive for local backups to an RS/6000, set the fixed block size to 1024 to avoid wasting tape. For example,

```
# chdev -l rmt0 -a block_size=1024
```

Summary

AIX and HP-UX support a wide variety of backup utilities, ranging from the proprietary to the universal. AIX's proprietary commands are **backup** and **restore**, based on **dump** and **restore**. **backup** can do a file system dump or it can back up individual files. **restore** can restore an entire file system or individual files as well. AIX writes to tape either fixed-length blocks or variable-length blocks. You can configure the system for one or the other by using SMIT or the **chdev** command. For network backups or when the application determines block size, it is best to set the block size to variable length. Instead of having the **mt** command, AIX has the **tctl** command, which is very similar to **mt**.

HP-UX's proprietary backup utilities are **fbackup** and **frecover**. These commands resemble **backup** and **restore** to an extent but are much more versatile, particularly in the area of graph files and network backups. To improve performance while writing to a cartridge tape, HP-UX supplies the **tcio** command.

Interplatform backups are possible using standard UNIX utilities like **tar**, **cpio**, **dd**, and **pax**, and the networking command **remsh**. This command needs the appropriate **.rhosts** entries in order to work properly. If you write to another system's tape drive, you must use a common command like **tar** if you want to eventually restore to the other system. But, for example, if you use **backup** to write to an HP-UX machine's tape drive, you can only restore your files to an AIX system. For best portability, use **pax**, the portable archive exchange command.

8. Printing

In discussing UNIX printing, we have to be careful about the terms we use to avoid confusion. The respective documentation for AIX and HP-UX often use similar terms but with different meanings. In this document the following terms will be used consistently:

- **`Local printer`**

  A printer that is physically connected to your computer.

- **`Remote printer`**

  A printer that is physically connected to another computer and accessed over a network

- **`Network-based printer`**

  A printer or plotter that is directly connected to the local area network

AIX

The examples in this section show connections to Hewlett-Packard printers. To make this possible the following file sets were loaded:

`bos.txt.hplj.fnt`	4.1.0.0	Fonts for Hewlett Packard Laser
`printers.hpJetDirect.attach`	4.1.3.0	Hewlett-Packard JetDirect
`printers.hplj-3.rte`	4.1.4.0	Hewlett-Packard LaserJet III
`printers.hplj-4.rte`	4.1.4.0	Hewlett-Packard LaserJet 4
`printers.hplj-4si.rte`	4.1.4.0	Hewlett-Packard LaserJet 4si

The AIX Queuing System

To print in AIX you use a proprietary queuing system that is not confined to printing alone. In other words, the queuing system accepts jobs other than print jobs. For purposes of this discussion, however, we will assume all jobs are print jobs.

You can submit print jobs with the **`enq`** command, the general queuing command, or you can use the **`qprt`**, **`lp`**, or **`lpr`** commands. AIX provides **`lp`** and **`lpr`** for purposes of compatibility with AT&T and BSD-style printing, but they do not represent completely different printing subsystems; they are simply front ends that do the same thing. All three print commands, **`qprt`**, **`lp`**, and **`lpr`**, call the **`enq`** command when invoked.

`enq` submits jobs to a queue, either one you specify or a default queue. In AIX-speak a queue is "an ordered list of requests for a specific device."[8] This device is known as a **`queue device`**, and is not the printer or even its device file. A queue device is actually a set of parameters for the print device. To illustrate this, let's look at a sample **`/etc/qconfig`** file. **`/etc/qconfig`**

[8]**InfoExplorer**, *Queuing System Overview for System Management*. HP-UX defines a queue as a directory used by the **lp spooler** to hold print jobs for each print destination until they can be printed. Thus the concept of queue is quite different.

is the master configuration file for the queuing system. It is an ASCII file consisting of stanzas that describe queues and their associated devices. For example, an entry might look like the following:

```
lp0:
     device = pdev0
pdev0:
     file = /dev/lp0
     header = never
     backend = /usr/lpd/piobe
```

Queue **`lp0`** is listed with an associated queue device of **`pdev0`**, which has three configuration parameters. *file* is the special device file of the printer. *header* specifies whether or not header pages are printed; in this case, no. *backend* is the program that actually sends jobs to the printer; in this case it is **`/usr/lpd/piobe`**.

The basic steps of the print process are:

1. **`enq`**, either directly or through the **`qprt`**, **`lp`**, or **`lpr`** commands, submits jobs to a queue as defined in **`/etc/qconfig`** and places them in the **`/var/spool/lpd/qdir`** directory.
2. The **`qdaemon`** awakens and places the print spool file in **`/var/spool/qdaemon`**. The **`qdaemon`**, which is started at boot time, tracks print requests and printer availability. When a printer becomes available, **`qdaemon`** submits the job to the printer backend.
3. The backend program sends the job to the printer and at the same time keeps status of the print job in the **`/var/spool/lpd/stat`** directory. This directory is consulted when you use one of the queue checking programs: **`qchk`**, **`lpstat`**, or **`enq -A`** commands.

Adding a Local Printer

1. Start SMIT:

   `# smit mkpq`

   You will see the following:

```
                           Add a Print Queue

Move cursor to desired item and press Enter. Use arrow keys to scroll.

  # ATTACHMENT TYPE      DESCRIPTION
    local                Printer Attached to Local Host
    remote               Printer Attached to Remote Host
    xstation             Printer Attached to Xstation
    ascii                Printer Attached to ASCII Terminal
    hpJetDirect          Network Printer (HP JetDirect)
    file                 File (in /dev directory)
    other                User Defined Backend
```

2. Select **`local`** attachment type. You will see the following:

```
                             Printer Type

Move cursor to desired item and press Enter.

  Bull
  Canon
  Dataproducts
  Hewlett-Packard
  IBM
  OKI
  Printronix
  QMS
  Texas Instruments
  Other (Select this if your printer type is not listed above)
```

3. Select the appropriate manufacturer. This example will use theHewlett-Packard 4Si, so in this case you would select **`Hewlett-Packard`** and then see the following and select **`hplj-4si`**:

```
                              Printer Type

Move cursor to desired item and press Enter.

  hplj-2    Hewlett-Packard LaserJet II
  hplj-3    Hewlett-Packard LaserJet III
  hplj-3si  Hewlett-Packard LaserJet IIISi
  hplj-4    Hewlett-Packard LaserJet 4,4M
  hplj-4si  Hewlett-Packard LaserJet 4Si
  Other (Select this if your printer type is not listed above)
```

4. From the **`Printer Interface`** menu select the appropriate interface: **`parallel`**, **`rs232`**, or **`rs422`**. Our example will use **`parallel`**.

5. From the **`Parent Adapter`** menu select the appropriate adapter. For parallel you should see only one choice:

   `ppa0 Available 00-00-0P Standard I/O Parallel Port Adapter`

   For serial you should see two choices:

   `sa0 Available 00-00-S1 Standard I/O Serial Port 1`

   `sa1 Available 00-00-S2 Standard I/O Serial Port 2`

6. If you are adding a parallel printer the next menu is the following:

```
                            Add a Print Queue

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Description                                   Hewlett-Packard LaserJ>

  Names of NEW print queues to add
     PCL                                       []
     PostScript                                []
     HP-GL/2                                   []

  Printer connection characteristics
*    PORT number                               [p]                    +
     Type of PARALLEL INTERFACE                [standard]             +
     Printer TIME OUT period (seconds)         [600]                  +#
     STATE to be configured at boot time        available             +
```

Since the **4Si** supports PCL, PostScript, and HP-GL/2 print data, you should enter a queue name for each. This example has three queues: **lpcl** (local PCL), **lps** (local PostScript), and **lhpgl** (local HPGL).

This procedure combines what in AIX 3.2.5 was two steps: create the **/dev/lp0** device file and create print queues for that device. The print queues are created with default attributes and can be viewed by checking the **/etc/qconfig** file, which in our example contains the following:

```
lpcl:
        device = lp0
lp0:
        file = /dev/lp0
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/piobe
lps:
        device = lp0
lp0:
        file = /dev/lp0
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/piobe
lhpgl:
        device = lp0
lp0:
        file = /dev/lp0
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/piobe
```

The queue **lpcl** was made the default queue and therefore appears first in **/etc/qconfig**. Unless you specify a print queue on the command line, all print jobs go to queue **lpcl**. Each queue has the same queue device because we are printing to the same printer. However, if we want to print in PostScript mode, we use the second queue; for HPGL, the second. For example,

For queue **`lps`**:

```
# enq -P lps file
# qprt -P lps file
# lp -d lps file
# lpr -P lps file
```

For queue **`lhpgl`**:

```
# enq -P lhpgl file
# qprt -P lhpgl file
# lp -d lhpgl file
# lpr -P lhpgl file
```

Printing to a Remote Printer

The basic process of printing to a remote printer is to

1. Attach a printer to the remote machine and add it as a local printer to the remote workstation.
2. Start the **`lpd`** daemon on the remote machine and enable access to it from your local workstation.
3. Create a remote queue on the local workstation.

Remote Workstation Setup

Attach a printer to the remote workstation in the same manner as described in **`Adding a Local Printer`**. Though you are setting up remote printing, the printer is a local printer to the remote workstation. The remote workstation with a printer attached is frequently called a print server; a local workstation printing to a print server is often called a print client. We will use that terminology here to help avoid some confusion. After you add the printer, do the following:

1. Start SMIT:

   ```
   # smit server
   ```

2. Select **`Add Print Access for a Remote Client`**.
3. In the **`Name of REMOTE CLIENT`** field, fill in the name or IP address of your print client.

   The terminology here is confusing, but try to remember: what is local to you is remote to the remote workstation. You are adding a remote host name to the print server's list of authorized print client workstations This list is the ***`/etc/hosts.lpd`*** *file..*

4. Return to the first menu (press the **`F3`** key twice if you are in ASCII SMIT). Select **`Start the Print Server Subsystem (lpd daemon)`**.
5. In the **`Start the Print Server Subsystem`** screen, choose **`both`** to start the **`lpd`** and put an entry into the **`/etc/inittab`** file for **`lpd`** to start upon boot.

Local Workstation Setup

Before you set up the local workstation, it is good to know the different ways a print job can be filtered. AIX 4 supports the following:

Standard processing	Filtering is done on the printer server. No processing done local workstation
Standard proessing with NFS access to server print queue attributes	Filtering is done locally but print queue attributes must exist on printer server. Print server must be AIX 4 system.
Local processing	All filtering is done locally.

To create a remote print queue:

1. Start SMIT:

   `# smit mkpq`

   You will see the following:

```
                            Add a Print Queue

Move cursor to desired item and press Enter. Use arrow keys to scroll.

   # ATTACHMENT TYPE      DESCRIPTION
     local                Printer Attached to Local Host
     remote               Printer Attached to Remote Host
     xstation             Printer Attached to Xstation
     ascii                Printer Attached to ASCII Terminal
     hpJetDirect          Network Printer (HP JetDirect)
     file                 File (in /dev directory)
     other                User Defined Backend
```

2. Select **`remote`**. You will then see:

```
                         Type of Remote Printing

Move cursor to desired item and press Enter.

  Standard processing
  Standard with NFS access to server print queue attributes
  Local filtering before sending to print server
```

   Given the descriptions above, select the appropriate filtering scheme. If you choose standard processing, you will see:

```
                    Add a Standard Remote Print Queue

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* Name of QUEUE to add                          []
* HOSTNAME of remote server                     []
* Name of QUEUE on remote server                []
  TYPE of print spooler on remote server         AIX Version 3 or 4     +
  DESCRIPTION of printer on remote server       []
```

For **`NAME of QUEUE to add`**, type in anything you want. It is a good idea, however, to use some kind of logical naming system, such as **`rlp0`** (remote line printer 0). For **`HOSTNAME of remote server`**, type the hostname of the print server. For **`Name of QUEUE on remote server`**, type the name of the print server's local queue, for example **`lp0`**. In the **`TYPE of print spooler on remote server`** you have the following choices:

- AIX Version 3 or 4
- BSD
- System V
- AIX Version 2 (RT PC)

These selections refer to what filters are used to translate remote queue status. If you are printing to an HP-UX machine, you would choose **`System V`**. This enables the filter to translate the **`lpstat`** command output.

After completion you will have an entry in **`/etc/qconfig`** for the remote printer. Consider the following sample entry that has a remote queue called **`rlp0`** which prints to print server **`hpbarr1.nsr.hp.com`** and uses two filters for AT&T machines:

```
rlp0:
        device = @hpbarr1
        up = TRUE
        host = hpbarr1.nsr.hp.com
        s_statfilter = /usr/lib/lpd/attshort
        l_statfilter = /usr/lib/lpd/attlong
        rq = lp0
@hpbarr1:
        backend = /usr/lib/lpd/rembak
```

The queue **`lp0`** on **`hpbarr1.nsr.hp.com`** is accessed from the RS/6000 via the **`rlp0`** queue. To print the **`/etc/hosts`** file you can type:

```
 # enq -P rlp0 /etc/hosts
```

If you choose to have filtering on locally, then SMIT will ask you what type of printer is attached to the remote system in order to set up the filtering mechanism. The following example shows the **`/etc/qconfig`** entry for a Hewlett-Packard 4Si printer in which the backend **`piorlfb`** runs to complete the print job:

```
rlp0:
        device = @hpbarr1
        host = hpbarr1.nsr.hp.com
        rq = lp0
        s_statfilter = /usr/lib/lpd/aixshort
        l_statfilter = /usr/lib/lpd/aixlong
@hpbarr1:
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/pio/etc/piorlfb -f !
```

Printing to a Network-based Printer

AIX supports network-based printing if the networked printer supports a JetDirect card. For other types of cards or network printers, you need to install the appropriate software that comes with the card or printer. To connect to a JetDirect printer,

1. Start SMIT:

   ```
   # smit mkpq
   ```

2. On the Add a Print Queue screen select **`hpJetDirect`**.

3. On the **`Printer Type`** screen select **`Hewlett-Packard`**. Another **`Printer Type Screen`** appears like this:

```
                              Printer Type

Move cursor to desired item and press Enter.

   hplj-2    Hewlett-Packard LaserJet II
   hplj-3    Hewlett-Packard LaserJet III
   hplj-3si  Hewlett-Packard LaserJet IIISi
   hplj-4    Hewlett-Packard LaserJet 4,4M
   hplj-4si  Hewlett-Packard LaserJet 4Si
   Other (Select this if your printer type is not listed above)
```

4. Select a printer. Our example uses **`hplj-4si`**.

   The next screen asks whether to make your machine a BOOTP/TFTP server. This is not necessary if such a server has already been created.

```
                            BOOTP/TFTP Server

 Move cursor to desired item and press Enter.

   1 Make this system a BOOTP/TFTP server
   2 Do NOT make this system a BOOTP/TFTP server
```

   Our example is to select **`2`**, in which case you get the following screen:

```
                            Add a Print Queue

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                            [Entry Fields]
  Description                                        Hewlett-Packard LaserJ>

  Names of NEW print queues to add
     PCL                                            []
     PostScript                                     []
     HP-GL/2                                        []

  Printer connection characteristics
*    HOSTNAME of the JetDirect Card                 []
```

5. Specify the hostname of the JetDirect printer and create queues for the various printer data types.

The following **`/etc/qconfig`** file shows an example for three queues, **`ps0`**, **`pcl0`**, and **`hpgl0`**:

```
ps0:
        device = hp@p2410l4si
hp@p2410l4si:
        file = /var/spool/lpd/pio/@local/dev/hp@p2410l4si#hpJetDirect
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/pio/etc/piojetd p2410l4si
pcl0:
        device = hp@p2410l4si
hp@p2410l4si:
        file = /var/spool/lpd/pio/@local/dev/hp@p2410l4si#hpJetDirect
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/pio/etc/piojetd p2410l4si
hpgl0:
        device = hp@p2410l4si
hp@p2410l4si:
        file = /var/spool/lpd/pio/@local/dev/hp@p2410l4si#hpJetDirect
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/pio/etc/piojetd p2410l4si
```

AIX Print Commands

As stated at the beginning of this chapter, you can submit print jobs with the **`enq`** command, the general queuing command, or you can use the **`qprt`**, **`lp`**, or **`lpr`** commands. AIX provides **`lp`** and **`lpr`** for purposes of compatibility with AT&T and BSD-style printing. Other commands include:

Submit print jobs	**Status print jobs**	**Cancel print jobs**
enq	enq -A	enq -x
qprt	qchk	qcan
lp	lpstat	lpq
lpr	lpq	lprm

Queue Management Commands

To bring down the printer but not its queue:

```
# qadm -D Queue
```

To bring down the printer and kill all current jobs:

```
# qadm -K Queue
```

To kill all current jobs:

```
# qadm -X Queue
```

To bring down the entire queuing system, allowing queued jobs to finish before doing so:

```
# qadm -G Queue
```

To bring the queuing system back up:

```
# qadm -U Queue
```

HP-UX

LP Spooler Tasks

Initializing the LP Spooler

Before you can use the LP spooler, you need to initialize it as follows:

1. Add at least one printer to the LP spooler.
2. Tell the LP spooler to accept print requests for this printer.
3. Tell the LP spooler to enable the printer for printing.
4. Turn on the LP spooler.
5. Note: If you use SAM to add a printer, SAM will do the above steps.

Specifying the Printer Model File to the LP Spooler

When you configure your printer into the LP spooler, you must identify the printer interface script to be used. There are printer interface scripts you can choose from in the **`/usr/lib/lp/model`** directory. This directory contains files corresponding to the models and names of all HP printers and plotters (plus some generic model files). The following table lists the names of the basic model files, the additional models to which they are linked, and the HP product numbers they support.

The **`/usr/sbin/lpadmin`** command will copy the identified model script to **`/etc/lp/interface/`*`printername`***. See the **`lpadmin man`** for information on the command options.

If you are configuring a non-HP printer to HP-UX, read the ASCII model files to identify the essential printer characteristics—such as whether your printer uses Printer Command Language (PCL) or PostScript. Also see the manual that came with your printer for more information on PCL language levels. For third-party printers that are not PostScript printers, use the model dumb; for non-PostScript plotters, use dumbplot.

Model Files and Corresponding Printers and Plotters

Model File	Intended Purpose
HPGL1	LP interface for HP7440A, HP7475A plotter; identical files: colorpro, hp7440a, hp7457a
HPGL2	LP interface for HP7550A, HP7596A, HP7570A plotter; identical files: hp7550a, hp7570a, hp7595a, hp7596a, draftpro
HPGL2.cent	LP interface for HP7550Plus, HP7550B plotters, and 7600 Series Electrostatic plotters when connected via parallel interface
PCL1	PCL level 1 model interface; identical files: hp2225a, hp2225d, hp2227a, hp2228a, hp2631g, hp3630a, paintjet, quietjet, thinkjet
PCL2	PCL level 2 model interface; identical files: hp2300-1100L, hp2300-840L, hp2560, hp2563a, hp2564b, hp2565a, hp2566b, hp2567b
PCL3	PCL level 3 model interface; identical files: deskjet, deskjet500, deskjet500C, deskjet550C, hp2235a, hp2276a, hp2932a, hp2934a, ruggedwriter
PCL4	PCL level 4 model interface; identical files: hp33447a, laserjet, hp5000fl100
hp33440a	Model file based on PCL level 4; identical files: hp2684a, hp2686a
PCL5	PCL level 5 model interface; identical files: hp5000c30, laserjet4Si, laserjetIIISi, laserjet4
deskjet1200C	LP interface based on PCL5, including support for language switching; identical file: deskjet1200C (this is the same file name as the model file), painjetXL300
hpC1208a	LP interface for HP C1208A, based on PCL5
dumb	LP interface for dumb line printer
dumbplot	LP interface for dumb plotter
postscript	LP interface for PostScript printer, for use on HP LaserJet IID, III, printers with HP 33439P LaserJet PostScript cartridge, as well as generic PostScript printers
rmodel	LP interface for remote printers

Adding a Local Printer to the LP Spooler

Adding a printer to the LP spooler differs from adding a printer to your system: adding a printer to the LP spooler involves configuring the LP spooler, whereas adding a printer to your system involves connecting the printer to your computer and configuring the needed drivers in the kernel. For information on the latter, refer to *Configuring HP-UX for Peripherals*.

The easiest way to add a local printer to the LP spooler is to run SAM. If you decide to use HP-UX commands instead, follow these steps:

1. Ensure that you have superuser capabilities.
2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. Add the printer to the LP spooler. For example:

   ```
   # lpadmin -pchk_printer -v/dev/lp -mhp2934a -g7
   ```

   See the **lpadmin man** page for details on the options.

4. Allow print requests to be accepted for the newly added printer.

   ```
   # accept chk_printer
   ```

5. Enable the newly added printer to process print requests. For example:

   ```
   # enable chk_printer
   ```

6. Restart the LP spooler:

   ```
   # lpsched
   ```

Adding a Remote Printer to the LP Spooler

The easiest way to add a printer to a remote system is to run SAM. If you decide to use HP-UX commands instead, follow the above steps in **Adding a Local Printer to the LP Spooler**, except replace Step 3 with the following:

If the remote printer is on an HP-UX system, enter:

```
# lpadmin -plocal_printer -v/dev/null -mrmodel -ormremote_machine \
-orpremote_dest -ocmrcmodel -osmrsmodel
```

If the remote printer is not on an HP-UX system, enter:

```
# lpadmin -plocal_printer -v/dev/null -mrmodel -ormremote_machine -
orpremote_dest -ocmrcmodel -osmrsmodel -ob3
```

See **lpadmin(1M)** for details on the options.

If your remote printer does not work, check if the remote printing daemon (**rlpdaemon**) is correctly running on the remote machine (that is, the host on which the physical printer resides):

1. Examine the file **/etc/inetd.conf** and look for the following line:

   ```
   # printer stream tcp nowait root /usr/sbin/rlpdaemon  rlpdaemon -i
   ```

If there is a **#** sign at the beginning of the line, the **`rlpdaemon`** line is commented out, preventing the printer from printing remotely. Edit the file **`/etc/inetd.conf`** to remove the **#** sign. Save the file.

2. Check **`/etc/services`** and look for:

   ```
   # printer 515/tcp spooler #remote print spooling
   ```

   If there is a **#** sign at the beginning of the line, it is commented out. Edit the file to remove the **#** sign in the first column and save the file.

3. Reconfigure the Internet daemon **`inetd`**, forcing it to reread the **`/etc/inetd.conf`** file:

   ```
   # inetd -c
   ```

4. Also, check entries in **`/var/adm/inetd.sec`** that restrict which systems can send remote print requests.

Adding a Network-Based Printer

A network-based printer is connected directly to the LAN, thus is not physically connected to any system. Network printers do not use device special files.

You can use SAM to add a network-based printer that uses the HP JetDirect Network Interface. The HP JetDirect software must be installed on your system and you must be prepared to provide SAM with the printer's node name (the name associated with an Internet address) and the local name that the LP spooler will use to refer to the printer. With HP JetDirect, printers can connect directly to the network. The printer uses a LAN connection and the HP JetDirect software transmits prints requests. For more information, see *HP JetDirect Network Interface Configuration Guide*.

If you do not use SAM, follow the instructions shipped with your printer or the network interface card for the printer.

Creating a Printer Class

You can make efficient use of multiple printers by creating a printer class. A printer class is a name you use to refer to a group of printers. Print requests can then be spooled to a single print queue and print requests will be printed by the first available printer in the class. Thus logjams on a particular printer are reduced or avoided. (Note that remote printers cannot belong to a printer class.)

You can use SAM to add a printer to a printer class when the printer is being added to the spooler; otherwise, you must use HP-UX commands. To use HP-UX commands, follow these steps after several printers have been added to the LP spooler:

1. Ensure that you have superuser capabilities.

2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. Create the printer class, specifying the printer you want to add to the class of printers. For example, to add a printer named **`laser1`** to the class of printers named **`laser`**, enter:

```
# lpadmin -plaser1 -claser
```

4. Only one printer can be added to a class at a time. If you have more than one printer to add, repeat this command.
5. The **lpadmin** command can add a printer to a new class, add a printer to an existing class, or move a printer from one class to another class (a printer can only belong to one class).
6. Allow print requests to be accepted for the newly added printer class. For example:

   ```
   # accept laser
   ```
7. Restart the LP spooler:

   ```
   # lpsched
   ```

Removing a Printer from the LP Spooler

You can use SAM or HP-UX commands to remove a printer from the LP spooler. If you use SAM, SAM asks for confirmation before removing the printer. If there are print jobs in the printer's queue, or if the printer is the system default destination, SAM's confirmation message will include that information. If you choose to remove a printer that has jobs in its queue, SAM cancels those jobs.

If you use HP-UX commands, follow these steps:

1. Ensure that you have superuser capabilities.
2. (Optional): Notify users that you are removing the printer from the system.
3. Remove the printer from the configuration file of any software application through which the device is accessed. (Refer to the documentation accompanying the software application for instructions.)
4. Stop the LP spooler:

   ```
   # lpshut
   ```
5. (Optional): Deny any further print requests for the printer. For example:

   ```
   # reject -r"Use alternate printer." laser1
   ```

   By doing this step, you can be assured that no new jobs will appear before you remove the printer. Users will see the message "Use alternate printer" when they direct requests to a rejected destination if the printer has not yet been removed. Once the printer has been removed and a user tries to send a request, they will see the message "Destination printer_name non-existent".
6. (Optional): Determine if there are any jobs in the printer's queue. For example:

   ```
   # lpstat -o laser1
   ```
7. (Optional): Disable the printer to be removed. For example:

   ```
   # disable -r"Printer laser1 is disabled." laser1
   ```

You would issue the above disable command if there are jobs in the printer's queue and you do not want to wait for them to print before removing the printer. Issuing the disable command shuts the printer down in an orderly manner. Note that you can also specify the **-c** option to the disable command to cancel all print requests for the printer.

8. (Optional): If there are no jobs in the printer's queue, go on to Step 9. If there are jobs, decide whether to move all pending print requests in the request directory to another printer request directory or to cancel any requests. For example, to move print requests:

   ```
   # lpmove laser1 laser2
   ```

   To cancel any requests:

   ```
   # lpcancel laser1
   ```

9. Remove the printer from the LP spooler. For example:

   ```
   # lpadmin -xlaser1
   ```

10. Restart the LP spooler:

    ```
    # lpsched
    ```

Removing a Printer from a Printer Class

SAM does not provide a way to you to remove a printer from a class. Instead, use HP-UX commands as follows:

1. Ensure that you have superuser capabilities.
2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. Remove the printer from the class. For example:

   ```
   # lpadmin -plaser1 -rclass
   ```

4. Restart the LP spooler:

   ```
   # lpsched
   ```

Removing a Printer Class

SAM does not provide a way to you to remove a printer class. Instead, use HP-UX commands as follows:

1. Ensure that you have superuser capabilities.
2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. (Optional): Deny any further print requests for the printer. For example:

   ```
   # reject -r"Use alternate printer." laser1
   ```

4. (Optional): Determine if there are any jobs in the printer's queue. For example:

```
# lpstat -o laser1
```

5. (Optional): Move all pending print requests in the request directory for the printer class to another printer or printer class. For example:

   ```
   # lpmove laser1 laser2
   ```

6. Remove the printer class. For example:

   ```
   # lpadmin -xlaser1
   ```

7. Restart the LP spooler:

   ```
   # lpsched
   ```

8. When you remove a printer class, the printers in the class are not removed--you can still use them as individual printers. If you remove all printers from a class, that printer class is automatically removed.

Stopping and Restarting the LP Spooler

Typically, the LP spooler is started during the boot process. (To change the boot-up procedure by not starting the scheduler, edit the file **`/etc/rc.config.d/lp`** and set the shell environment variable **`LP`** to zero.)

The spooler must be stopped whenever the spooling system needs to be modified (such as when adding or removing a printer) and subsequently restarted after the modification has been made. You can use either SAM or HP-UX commands to stop or start the LP spooler.

If you use HP-UX commands to stop the LP spooler, follow these steps:

1. Ensure that you have superuser capabilities.
2. Check if there are any requests printing or being sent to a remote printer (it is best to wait until there are no requests printing before stopping the LP spooler).

   ```
   # lpstat -o -i
   ```

   In the above command, the **`-i`** option inhibits the reporting of remote requests (that is, **`lpstat`** will only show local requests).

3. Stop the LP spooler:

   ```
   # lpshut
   ```

   All requests printing when **`lpshut`** is executed will be stopped, but will remain in the print queues.

4. Restart the LP spooler:

   ```
   # lpsched
   ```

5. When the spooler is restarted, the requests in the print queue will be completely reprinted regardless of how much of the request was printed prior to the shutdown.

Other Printing Tasks

Controlling the Flow of Print Requests

If you have superuser capabilities, you can use SAM or the HP-UX commands accept and reject to control the flow of print requests to the queues of named printers or printer classes. You can issue individual **`accept`** or **`reject`** commands for each printer or issue one command separating each printer by blank spaces.

If you use HP-UX commands to allow print requests to be sent to a printer or to a printer class, use the **`accept`** command. For example:

```
# accept laser1 jet2 lj
```

Use the **`reject`** command to temporarily prevent print requests from being sent to a printer or printer class. For example, to reject the **`lj`** class, enter:

```
# reject lj
```

If the **`reject`** command is executed on a printer class, but not on members of the class, users can still specify a specific printer (not the class) in subsequent print requests until an **`accept`** command on the class is re-issued.

If, however, you execute reject for all individual printers in a class, but not for the class itself, the print requests will remain in the class request directory until at least one of the printers in the class is permitted to process print requests by the **`accept`** command.

Enabling or Disabling a Printer

You can use SAM or the HP-UX commands enable and disable to activate or deactivate a printer for printing. You do not need superuser capabilities for these commands.

You can issue individual **`enable`** and **`disable`** commands for each printer or issue one command separating each printer by blank spaces. For example:

```
# enable laser1 laser2 laser3
```

You can enable or disable individual printers only, not printer classes. By default, any requests printing when a printer is disabled are reprinted in their entirety when the printer is re-activated. A printer that has been disabled can still accept new print requests to be printed at a later time unless it has been prevented from doing so by the **`reject`** command.

Controlling the Order of Printing

Each printer has two priority attributes:

- Fence priority
- Request priority

A printer's fence priority is used to determine which print requests get printed—only requests with priorities equal to or greater than the printer's fence priority get printed. You can assign the fence priority by using SAM or HP-UX commands.

To use HP-UX commands, follow these steps:

1. Ensure that you have superuser capabilities.

2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. Set the printer's fence priority (use a value from 0 to 7). For example:

   ```
   # lpfence myprinter 5
   ```

4. Restart the LP spooler:

   ```
   # lpsched
   ```

A print request has a request priority that is either assigned with the **-p** option of the **lp** command or is automatically assigned the printer's default request priority. You can change a print request's priority by using the **lpalt** command. A printer's default request priority can be set using the **lpadmin** command (SAM allows a default request priority other than zero to be set when a printer is added, but cannot change a printer's default request priority).

To change the default request priority, follow these steps:

1. Ensure that you have superuser capabilities.

2. Stop the LP spooler:

   ```
   # lpshut
   ```

3. Change the priority. For example:

   ```
   # lpadmin -pmyprinter -g7
   ```

   If you do not specify the **-g** option, the default request priority is set to zero.

4. Restart the LP spooler:

   ```
   # lpsched
   ```

If multiple print requests are waiting to be printed on a specific printer and all have priorities high enough to print, the printer will print the next print request with the highest priority. If more than one print request has the same priority, print requests with that priority will print in the order they were received by the LP spooler.

Setting Up the LP Spooler Using SAM

Adding a Local Printer

1. Physically connect the printer(s) to your system. Refer to the instructions shipped with your printer. You should always shut down your system and turn off the power when you are changing the hardware configuration of your system.

2. Gather the following information:

   - The name you are giving to this printer or plotter. Printer names can be up to 14 characters in length, and the characters must be alphanumeric (A-Z, a-z, 0-9) or an underscore (_).

   - The name of the device file that the printer or plotter will use. SAM creates the device file for you. SAM uses the default device file named **lp_printer-name**.

You can override the default device file name by specifying your device file name when filling in the printer information.

- The model script from the **`/usr/spool/lp/model`** directory, for example, laserjetIIISi for an HP LaserJet IIISi.
- The print request priority for this printer. The default is zero (0).
- The class to which the printer or plotter will be added (optional). Printer class names can be up to 14 characters in length, and the characters must be from the set (A-Z, a-z, 0-9). The underscore (_) character is allowed in printer class names.
- In addition, decide whether to make this device your system's default printer.

3. Run SAM.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Highlight **`LP Spooler`** and activate the **`Open`** control button.
6. Choose **`Printers and Plotters`**.
7. Choose **`Add Local Printer/Plotter >`** and the menu item associated with the printer interface type from the **`Actions`** menu.

   NOTE: The printer driver must be part of the kernel to add the printer to the **`lp`** spooler. If the printer driver is not currently configured into the kernel, SAM prompts you to add the driver(s) and reboot the system.
8. Highlight the interface to which you connected the printer and fill in and additional information (port number or bus address) and activate the **`OK`** control button. If an interface entry is not listed, activate the **`Diagnose Missing Card`** control button.
9. Fill in the printer interface dialog box fields, choose from the menu button values, and turn on and off check box values.
10. Activate the **`OK`** control button.

Adding a Remote Printer

1. Ensure that the remote system has the printer installed and configured into the remote system's line printer spooler system. Gather the following information:
2. The name you are giving to this printer or plotter.
   - Whether you wish to make this device your system's default printer.
   - The name of the remote system to which the printer or plotter is attached.
   - The name of the remote printer or plotter.
   - The "cancel" model on the remote system (optional).
   - The "status" model on the remote system (optional).
   - Whether you wish to allow any user to cancel any printing request.

- Whether the remote printer is on a system using BSD (Berkeley Software Distribution) UNIX. Using BSD disables any **`lp -oparm`** options. BSD systems do not understand the **`-o`** option.

3. Run SAM
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Highlight **`LP Spooler`** and activate the **`Open`** control button.
6. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
7. Choose **`Add Remote Printer/Plotter`** and the menu item associated with the printer interface type from the **`Actions`** menu.
8. Fill in the printer interface dialog box fields and turn on off check box values.
9. Activate the **`OK`** control button.

To configure a remote printer into your **`lp`** spooler, you must be able to access the system with the printer via a local area network (LAN). The process of adding a remote printer is similar to that of adding a local printer, though you will need to supply SAM with some slightly different information.

Remote printers cannot be members of a printer class.

Adding a Network-Based Printer

To add a network-based printer or plotter using SAM:

1. Ensure that the printer is connected to the network according to the installation instructions shipped with the network-based printer or the network interface card for the printer.
2. Gather the following information:
   - The name you are giving to this printer or plotter.
   - The printer node name.
   - The model or interface that the printer will use.
   - The link-level address of the network card installed in the printer.
   - The TCP-IP protocol printer requires an Internet Protocol (IP) address.
   - The priority for this printer.
   - The class to which the printer or plotter will be add.
   - In addition, decide whether you wish to make this device your system’s default printer.
3. Run SAM
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Highlight **`LP Spooler`** and activate the **`Open`** control button.

6. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
7. Choose **`Add Network-Based printer`** then **`Add Printer/Plotter Connected to HP JetDirect...`** from the **`Actions`** menu.
8. Fill in the printer interface dialog box fields and turn on and off check box values.
9. Activate the **`OK`** control button.

Starting and Stopping the LP Spooler

To start the **`lp`** spooler:

1. Run SAM.
2. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
3. Highlight **`LP Spooler`** and activate the **`Open`** control button.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Choose **`Start Print Spooler`** from the **`Actions`** menu.

To stop the **`LP`** spooler:

1. Run SAM.
2. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
3. Highlight **`LP Spooler`** and activate the **`Open`** control button.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Choose **`Shut Print Spooler`** from the **`Actions`** menu.

Determining the Status of the LP Spooler

To determine the status of the **`lp`** spooler:

1. Run SAM.
2. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
3. Highlight **`LP Spooler`** and activate the **`Open`** control button.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. The status area of the object list will display the status of the scheduler as **`Scheduler: RUNNING`** or **`Scheduler: STOPPED`**.

Disabling a Printer

1. Run SAM.
2. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
3. Highlight **`LP Spooler`** and activate the **`Open`** control button.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.

5. Highlight the printer you want to disable in the object list.
6. Choose **`Disable`** from the **`Actions`** menu.

NOTE: When you use SAM to "enable" or "disable" a printer, SAM performs both the accept/reject operation and the enable/disable operation. If you wish to "disable" a printer but still accept requests for that printer (letting them accumulate in the request directory for the printer), you must use the HP-UX commands method to disable the printer.

Enabling a Printer

To enable a printer using SAM:

1. Run SAM.
2. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
3. Highlight **`LP Spooler`** and activate the **`Open`** control button.
4. Highlight **`Printers and Plotters`** and activate the **`Open`** control button.
5. Highlight the printer you want to enable in the object list.
6. Choose **`Enable`** from the **`Actions`** menu.

Interplatform Printing

AIX can print to an HP-UX print server and vice versa. For this to work you must verify the following:

1. Basic network connectivity between the workstations
2. Each print server has the appropriate entries in either **`/etc/hosts.equiv`** or **`/etc/hosts.lpd`**.
3. For the queue of the AIX client printing to an HP-UX print server, use the following for filters in **`/etc/qconfig`**:

   `/usr/lpd/attshort`

   `/usr/lpd/attlong`
4. For printing from and HP-UX client to an AIX print server, be sure to choose the BSD option when adding a remote printer using SAM.

9. Configuring TCP/IP

AIX

Note: Beginning with AIX version 4 the InetServ ODM object class no longer exists. Therefore making changes to **`/etc/inetd.conf`** and **`/etc/services`** no longer requires the **`inetimp`** command. In fact, the **`inetimp`**, **`inetexp`**, **`inetserv`**, and **`notinet`** commands no longer exist.

On most UNIX systems you configure TCP/IP with commands such as **`ifconfig`** and **`route`**, and then make your configuration permanent by editing startup files. You can do this with AIX, but it is best to use SMIT to do all your configuration. Doing so ensures the information in **`/etc/rc.net`** and the ODM is updated.

Basic TCP/IP Configuration

1. Start SMIT:

   ```
   # smit mktcpip
   ```

2. On the **`Available Network Interfaces`** screen, select the appropriate interface. Upon bootup **`cfgmgr`** recognizes the network card and configures it into the ODM database.

3. On the **`Minimum Configuration & Startup`** menu, fill in, as a minimum, the **`HOSTNAME`**, **`Internet ADDRESS`** fields. Also specify whether or not you want to start TCP/IP now or at bootup in the **`START Now`** field.

4. Additional parameters that can be configured with this screen are network mask, name service, gateway, and cable types. If you leave the **`Network MASK`** field blank, AIX will provide you with a default value based upon the class of the system's IP address. For the **`CABLE Type`** field, **`bnc`** specifies thin cable, **`dix`** thick, **`tp`** twisted pair.

Example

```
                     Minimum Configuration & Startup

To Delete existing configuration data, please use Further Configuration
menus


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* HOSTNAME                                           [a2410bjv]
* Internet ADDRESS (dotted decimal)                  [15.24.48.58]
  Network MASK (dotted decimal)                      [255.255.248.0]
* Network INTERFACE                                   en0
  NAMESERVER
           Internet ADDRESS (dotted decimal)         [15.41.144.101]
           DOMAIN Name                               [nsr.hp.com]
  Default GATEWAY Address                            [15.24.55.253]
  (dotted decimal or symbolic name)
  Your CABLE Type                                     N/A             +
  START Now                                           no              +
```

Setting the Hostname

To set or reset your host’s name,

1. Start SMIT:

   ```
   # smit hostname
   ```

2. Choose the **`Set the Hostname`** menu item on the **`Hostname`** screen.
3. Fill in the new hostname in the **`HOSTNAME`** field.

Adding a Route

1. Start SMIT:

   ```
   # smit mkroute
   ```

2. Fill in the values for **`DESTINATION Address`** and **`Default GATEWAY Address`**. For **`Destination Type`** you have a choice between **`net`** and **`host`**. AIX provides a default value of **`1`** for the **`METRIC`** field.

```
                            Add Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                           [Entry Fields]
  Destination TYPE                                     net              +
* DESTINATION Address                                 []
  (dotted decimal or symbolic name)
* Default GATEWAY Address                             []
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway)     [1]               #
```

Removing a Route

Removing a route in SMIT looks much as the same as adding a route. Just type in the following and fill in the appropriate values:

```
 # smit rmroute
```

Flushing the Routing Table

To flush the routing table,

1. Start SMIT:

   ```
   # smit fshrttbl
   ```

2. This produces the following:

```
                           Flush Routing Table

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
  Flush Routing Table in the Current Running System   yes               +
  Flush Routing Table in the Configuration Data Base  no                +
   (effective in the next system restart)
```

If you want to flush the routing table temporarily but keep the routing information in the database, accept the defaults. If you want to clear the ODM of routing information, select *yes* for **`Flush Routing Table in the Configuration Data Base`**.

Changing Network Card Configuration

To alter the configuration of a network card, do the following:

1. Start SMIT:

   ```
   # smit chinet
   ```

2. Select the appropriate interface in the **`Available Network Interfaces`** menu.
3. Make the appropriate changes as needed. If you leave the **`BROADCAST ADDRESS`** field blank, AIX will provide a default based on the subnet mask.

Example

```
                Change / Show a Standard Ethernet Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                            [Entry Fields]
  Network Interface Name                                  en0
  INTERNET ADDRESS (dotted decimal)                      [15.24.48.58]
  Network MASK (hexadecimal or dotted decimal)           [255.255.248.0]
  Current STATE                                           up              +
  Use Address Resolution Protocol (ARP)?                  yes             +
  BROADCAST ADDRESS (dotted decimal)                     []
```

Removing Network Card Configuration

There is no SMIT fastpath option to removing an interface. Instead,

1. Start SMIT:

   `# smit inet`

2. Select the **`Remove a Network Interface`** option.

3. Select the appropriate interface.

CAUTION: SMIT removes the interface without prompting!

Managing Name Servers

To edit **`/etc/resolv.conf`**:

1. Start SMIT:

   `# smit resolv.conf`

2. This produces the following:

```
                   Domain Nameserver (/etc/resolv.conf)

Move cursor to desired item and press Enter.

  Start Using the Nameserver
  List All Nameservers
  Add a Nameserver
  Remove a Nameserver
  Stop Using a Nameserver
  ------------------------------
  Set / Show the Domain
  Remove the Domain
  Set / Show the Domain Search List
  Remove the Domain Search List
```

Use this as a means to edit the **`/etc/resolv.conf`** file or use a text editor to do so.

Adding Entries to /etc/hosts

1. Start SMIT:

   `# smit hostent`

2. This produces the following:

```
                          Hosts Table (/etc/hosts)

Move cursor to desired item and press Enter.

  List All Hosts
  Add a Host
  Change / Show Characteristics of a Host
  Remove a Host
```

Use this as a means to edit the **`/etc/hosts`** file or use a text editor to do so.

Editing /etc/inetd.conf

To edit **`/etc/inetd.conf`** and ensure the ODM database is updated:

1. Start SMIT:

   `# smit inetdconf`

2. This produces the following:

```
                              inetd Subservers

Move cursor to desired item and press Enter.

  List All inetd Subservers
  Add an inetd Subserver
  Change / Show Characteristics of an inetd Subserver
  Remove an inetd Subserver
```

Use the menu selections to add or remove entries to the **`/etc/inetd.conf`** and update the ODM. If, however, you edit **`/etc/inetd.conf`** with a text editor you must use the **`refresh -s inetd`** command to update the new information.

Editing the /etc/services File

You can use SMIT to edit **`/etc/services`**.

1. Start SMIT:

   `# smit inetserv`

2. This produces the following:

```
                        Services (/etc/services)

Move cursor to desired item and press Enter.

  List All Services
  Add a Service
  Change / Show Characteristics of a Service
  Remove a Service
```

Follow the menus to make the changes you desire. You can also use a text editor to change the **`/etc/services`** file.

Editing /etc/hosts.equiv

To create entries to or modify **`/etc/hosts.equiv`** using SMIT:

1. Start SMIT:

   ```
   # smit hostsequiv
   ```

2. This produces the following:

```
                      Host Access (/etc/host.equiv)

Move cursor to desired item and press Enter.

  List All Remote Hosts
  Add a Remote Host
  Remove a Remote Host
```

Follow the menus to make the changes you desire.

Editing /etc/ftpusers

To edit **`/etc/ftpusers`**:

1. Start SMIT:

   ```
   # smit ftpusers
   ```

2. This produces the following:

```
          Restrict File Transfer Program Users (/etc/ftpusers)

Move cursor to desired item and press Enter.

  Show All Restricted Users
  Add a Restricted User
  Remove a Restricted User
```

Managing Other Services

1. Start SMIT:

   # smit otherserv

2. This produces the following:

```
                         Other Available Services

Move cursor to desired item and press Enter.

  Super Daemon (inetd)
  syslogd Subsystem
  routed Subsystem
  gated Subsystem
  named Subsystem
  rwhod Subsystem
  timed Subsystem
  portmap Subsystem (information only)
  snmpd Subsystem
  dhcpsd Subsystem
  dhcpcd Subsystem
  dhcprd Subsystem
```

Using BSD Style rc Configuration

The AIX style of TCP configuration is to use the ODM configuration information. Upon bootup the **/etc/rc.net** file runs methods to configure the network card and set the hostname, default gateway, and routes based upon the ODM database. If you would prefer not to use the ODM to do this, you can elect to have a BSD style of TCP configuration. BSD style uses the **ifconfig** command and reads **/etc/rc.bsdnet** to configure the network card. To elect BSD style,

1. Start SMIT:

   # smit setbootup_option

2. Choose *yes* and press enter at the following screen:

```
                    Select BSD style rc Configuration

        Please answer yes if you want BSD style rc configuration.
                            The default is no.

    Default style configuration uses the data in the ODM database and
                  uses the file /etc/rc.net to define,
             load, and configure a corresponding interface.

   BSD style configuration uses the traditional ifconfig command and it
 uses the file /etc/rc.bsdnet to configure the corresponding interface.

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
  Use BSD Style rc Configuration               no                     +
```

A caveat about the above procedure: after selecting *yes* you will not have System Resource Controller (SRC) support. In other words, you can't use commands like **`refresh -s inetd`**. If you want flat file configuration *and* SRC support, uncomment the commands in **`/etc/rc.net`** under the heading **`Traditional Configuration`**. Below is that section in commented form:

```
##################################################################
# Part II - Traditional Configuration.
##################################################################
# An alternative method for bringing up all the default interfaces
# is to specify explicitly which interfaces to configure using the
# ifconfig command.  Ifconfig requires the configuration information
# be specified on the command line.  Ifconfig will not update the
# information kept in the ODM configuration database.
#
# Valid network interfaces are:
# lo=local loopback, en=standard ethernet, et=802.3 ethernet
# sl=serial line IP, tr=802.5 token ring, xs=X.25
#
# e.g., en0 denotes standard ethernet network interface, unit zero.
#
# Below are examples of how you could bring up each interface using
# ifconfig.  Since you can specify either a hostname or a dotted
# decimal address to set the interface address, it is convenient to
# set the hostname at this point and use it for the address of
# an interface, as shown below:
#
#/bin/hostname robo.austin.ibm.com      >>$LOGFILE 2>&1
#
# (Remember that if you have more than one interface,,
# you'll want to have a different IP address for each one.
# Below, xx.xx.xx.xx stands for the internet address for the
# given interface).
#
#/usr/sbin/ifconfig lo0 inet loopback    up >>$LOGFILE 2>&1
#/usr/sbin/ifconfig en0 inet 'hostname'  up >>$LOGFILE 2>&1
#/usr/sbin/ifconfig et0 inet xx.xx.xx.xx  up >>$LOGFILE 2>&1
#/usr/sbin/ifconfig tr0 inet xx.xx.xx.xx  up >>$LOGFILE 2>&1
#/usr/sbin/ifconfig sl0 inet xx.xx.xx.xx  up >>$LOGFILE 2>&1
#/usr/sbin/ifconfig xs0 inet xx.xx.xx.xx  up >>$LOGFILE 2>&1
```

```
#
#
# Now we set any static routes.
#
# /usr/sbin/route add 0 gateway                 >>$LOGFILE 2>&1
# /usr/sbin/route add 192.9.201.0 gateway       >>$LOGFILE 2>&1
```

HP-UX

The set_parms Script

On pre-installed or newly installed systems the startup script **`/etc/rc`** runs **`/sbin/set_parms`** to interactively set up a basic network configuration. The parameters that are set are:

- Host name
- Time zone
- IP address
- Subnet mask
- Default gateway
- Domain name server (DNS server)
- Network Information Service (NIS)
- Root password

You can reconfigure these basic network configurations later by starting **`/sbin/set_parms`** manually:

```
# set_parms initial
```

Networking Startup Files

When the system boots to run-level 2 or higher, the script **`/sbin/init.d/net`** starts. It fetches the configurable parameters from the file **`/etc/rc.config.d/netconf`** and executes the following commands:

`ifconfig`	Sets the IP address, subnet mask, and local loopback interface
`lanconfig`	Sets the encapsulation method
`route`	Configures the routing table

See the **`man`** pages for these commands for detailed information.

Editing the /etc/hosts File

You can use any text editor to edit the **`/etc/hosts`** file. If you are not running BIND or NIS, you can use SAM. SAM also configures **`/etc/hosts`** when adding a new LAN card. To use SAM,

1. Start SAM

```
# sam
```

2. Select **Networking and Communications**.

3. Select **Internet Addresses**.

4. Select **Actions->Add . . .**

5. Fill in the **Remote System Name** and **Internet Address** fields on the menu and select **Apply**.

The Name Service Switch

The Name Service Switch determines where your system will look for the information that is traditionally stored in the following files:

```
/etc/hosts
/etc/protocols
/etc/services
/etc/networks
/etc/netgroup
/etc/rpc
```

For all types of information except host information, you can configure your system to use NIS (one of the NFS Services), the local **/etc** file, or both, in any order. For host information, you can configure your system to use BIND (DNS), NIS, the **/etc/hosts** file, or any combination of the three, in any order. The default Name Service Switch configuration is adequate for most installations, so you probably do not have to change it.

NOTE: Configuring the Name Service Switch is a separate task from configuring the name services themselves. You must also configure the name services before you can use them. The Name Service Switch just determines which name services are queried and in what order. You can use SAM to configure the Name Service Switch. See the chapter on DNS in this book for more information about the Name Service Switch.

Choosing a Name Service

HP-UX provides three ways of translating host names to IP addresses or IP addresses to host names:

- The **/etc/hosts** file, a simple ASCII file that is searched sequentially.
- BIND (Berkeley Internet Name Domain), which is Berkeley's implementation of the Domain Name System (DNS).
- NIS (Network Information Service), one of the NFS Services. (NIS used to be called "Yellow Pages".)

By configuring the Name Service Switch, you can use these name services in any order you choose.

If you have a large network, or you need to connect to Internet hosts outside your local network, use BIND as your primary name service. When you use BIND, you administer a central database containing only the hosts on your local network, and you have access to the databases on all the other hosts on the Internet.

If you have a large network and little need for Internet connectivity, you can use NIS as your primary name service. The NIS hosts database is administered centrally on one of your hosts, but it must contain the names and IP addresses of all the other hosts in your network. For information on NIS, see Installing and Administering NFS Services.

If you have a small network and little need for Internet connectivity, you can use the **`/etc/hosts`** file as your primary name service. Each host in your network needs a copy of the **`/etc/hosts`** file containing the names and addresses of all the other hosts in your network.

If you choose to use BIND or NIS as your primary name service, you still need to configure a minimal **`/etc/hosts`** file so that your host can boot if BIND or NIS is not available.

Configuring Routes

1. If you use only one gateway to reach all systems on other parts of the network, configure a default gateway.

   You can use SAM to configure a default gateway, or if you are not using SAM, issue the following command:

   ```
   # /usr/sbin/route add default gateway_address 1
   ```

   where ***`gateway_address`*** is the IP address of the gateway host. Then, set the following environment variables in the **`/etc/rc.config.d/netconf`** file:

   ```
   ROUTE_DESTINATION[0]="default"
   ROUTE_GATEWAY[0]="gateway_address"
   ROUTE_COUNT[0]="1"
   ```

   If the default gateway is your own host, set the `ROUTE_COUNT` variable to 0. Otherwise, set it to 1.

2. If your host is a gateway, configure the destination networks that can be reached from its network interfaces. Issue the following command for each network interface on your host:

   ```
   # /usr/sbin/route add net destination IP_address
   ```

   where ***`destination`*** is a network address reachable by your host, and ***`IP_address`*** is the address of the network interface. Then, create a new set of routing variables in the **`/etc/rc.config.d/netconf`** file for each network interface.

   Whenever you create a new set of variables, increment the number in square brackets, as in the following example:

   ```
   ROUTE_DESTINATION[1]="15.13.131.0"
   ROUTE_GATEWAY[1]="15.13.131.213"
   ROUTE_COUNT[1]="0"
   ```

3. If you will not be using **`gated`**, configure routes to all the networks you need to reach. Type the following command for each network you need to reach from your host:

```
/usr/sbin/route add net network_address gateway_address
```

Then, create a new set of routing variables in the **/etc/rc.config.d/netconf** file for each new route. Whenever you create a new set of variables, increment the number in square brackets.

```
ROUTE_DESTINATION[n]="network_address"
ROUTE_GATEWAY[n]="gateway_address"
ROUTE_COUNT[n]="1"
```

If `ROUTE_GATEWAY[n]` is your own host, set `ROUTE_COUNT[n]` to 0. Otherwise, set it to 1.

4. Type the following command to verify the routes you have configured:

   ```
   # /usr/bin/netstat -r
   ```

   For more information on static routing, type **man 1M route** or **man 7 routing** at the HP-UX prompt.

To Set the Default Gateway Using SAM

1. Start SAM

   ```
   # sam
   ```

2. Select **Networking and Communications**.
3. Select **Internet Addresses**.
4. Choose **Modify Default Gateway. . .**
5. Fill in the **Default Gateway Internet Address** and **Default Gateway Name** fields and Choose **OK**.

Changing a Host's IP Address

When you use SAM to change a host's IP address, SAM does not perform all these steps. For example, SAM does not update BIND or NIS databases.

1. Change the host's IP address in the **/etc/hosts** file.
2. Change the `IP_ADDRESS[n]` variable in the **/etc/rc.config.d/netconf** file to the new IP address.
3. If the host is on a network that uses BIND, change the host's IP address in the data files of the authoritative name servers. If the host is on a network that uses NIS, change its IP address in the **/etc/hosts** file on the NIS master server, and issue the following commands to regenerate the hosts database and push it out to the NIS slave servers:

   ```
   # cd var/yp
   # /usr/ccs/bin/make hosts
   ```

4. If the host is moving to a different subnet, change the `ROUTE_DESTINATION`, `ROUTE_GATEWAY`, and `BROADCAST_ADDRESS[n]` variables in **/etc/rc.config.d/netconf**.

If the host is moving to a nework that uses a different subnet mask, change the `SUBNET_MASK[n]` variable in **`/etc/rc.config.d/netconf`**.

5. If the host is moving to a different network, you may have to configure new routes for it.
6. If the host is on a network that uses **`gated`**, change its IP address on all the **`gated`** routers.
7. If the host is a BOOTP client, change its IP address in the **`/etc/bootptab`** file on the BOOTP server. If the host is a BOOTP server, and a BOOTP relay agent is configured to relay boot requests to the host, change the host's IP address in the **`/etc/bootptab`** file on the BOOTP relay agent.
8. If the host is an NTP server, change its IP address in the **`/etc/ntp.conf`** file on NTP clients. If the host is an NTP client and is moving to another network, you might have to configure a different NTP server in its **`/etc/ntp.conf`** file.
9. Reboot the host.

Configuring inetd

The **`inetd`** daemon is always started as part of the boot process by the startup script **`/sbin/init.d/inetd`**. In addition to the **`/etc/inetd.conf`** configuration file, you can configure an optional security file called **`/var/adm/inetd.sec`**, which restricts access to the services started by **`inetd`**.

1. Make sure the following lines exist in **`/etc/inetd.conf`**. If any of the lines starts with a pound sign (#), remove the sharp sign to enable the service.

```
ftp    stream tcp nowait root /usr/lbin/ftpd    ftpd -l
telnet stream tcp nowait root /usr/lbin/telnetd telnetd
tftp   dgram  udp wait   root /usr/lbin/tftpd   tftpd
bootps dgram  udp wait   root /usr/lbin/bootpd  bootpd
finger stream tcp nowait bin  /usr/lbin/fingerd fingerd
login  stream tcp nowait root /usr/lbin/rlogind rlogind
shell  stream tcp nowait root /usr/lbin/remshd  remshd
exec   stream tcp nowait root /usr/lbin/rexecd  rexecd
```

   To disable any of these services, comment out the line by typing a pound sign (#) as the first character on the line.

2. If you made any changes to **`/etc/inetd.conf`**, type the following command to force inetd to read its configuration file:

```
# /usr/sbin/inetd -c
```

3. Make sure **`/etc/inetd.conf`** is owned by user root and group other, and make sure its permissions are set to 0444 (-r--r--r--).

For more information, type **`man 4 inetd.conf`** or **`man 1M inetd`**.

Editing the /var/adm/inetd.sec File

The **`/var/adm/inetd.sec`** file is a security file that **`inetd`** reads to determine which remote hosts are allowed access to the services on your host. The **`inetd.sec`** file is optional; you do not need it to run the Internet Services.

You can use either a text editor or SAM to edit the **`inetd.sec`** file.

1. If the **`/var/adm/inetd.sec`** file does not exist on your host, copy **`/usr/newconfig/var/adm/inetd.sec`** to **`/var/adm/inetd.sec`**.

   Create one line in inetd.sec for each service to which you want to restrict access. Do not create more than one line for any service. Each line in the **`/var/adm/inetd.sec`** file has the following syntax:

   ```
   service_name {allow | deny} host_specifier [host_specifier...]
   ```

   where **`service_name`** is the first field in an entry in the **`/etc/inetd.conf`** file, and **`host_specifier`** is a host name, IP address, IP address range, or the wildcard character (*).

2. Make sure the **`/var/adm/inetd.sec`** file is owned by user root and group other, and make sure its permissions are set to 0444 (-r--r--r--).

Following are some example lines from an **`inetd.sec`** file:

```
login allow 10.*
shell deny vandal hun
tftp deny *
```

The first example allows access to **`rlogin`** from any IP address beginning with 10. The second example denies access to **`remsh`** and **`rcp`** from hosts **`vandal`** and **`hun`**. The third example denies everyone access to **`tftp`**. Only the services configured in **`/etc/inetd.conf`** can be configured in **`/var/adm/inetd.sec`**.

Enabling bootp and tftp

1. Start SAM:

   ```
   # sam
   ```

2. Select **`Networking and Communications`**.
3. Select **`Network Services`**.
4. Select **`Bootp`**.
5. Choose **`Actions->Enable`**.
6. Select **`TFTP`**.
7. Choose **`Actions->Enable`**.

Configuring rwhod

The **`rwhod`** daemon checks the state of your host and generates status messages, which it broadcasts on the network every 180 seconds. It also listens for status messages broadcast by

rwhod daemons on remote hosts, and it records these messages in a database of files in **/var/spool/rwho**. The files are named **whod.hostname**, where hostname is the name of the remote host from which the status information came. The status messages are displayed when users issue the **rwho** or **ruptime** command.

1. In the **/etc/rc.config.d/netdaemons** file, set the `RWHOD` variable to 1.
2. Issue the following command to start the **rwhod** daemon:

   ```
   # /sbin/init.d/rwhod start
   ```

Status information collected by **rwhod** for the local host and from each remote host includes the following:

- System load average.
- Host name as returned by **gethostbyname**.
- Users logged in.
- Time of last activity for logged-in users.

Because UDP (User Datagram protocol) broadcasts do not go through gateways, **rwho** and **ruptime** do not report status for hosts that can be reached only through a gateway.

Configuring Logging for the Internet Services

syslogd

The Internet daemons and servers log informational and error messages through **syslog**. You can monitor these messages by running **syslogd**. You can determine the type and extent of monitoring through **syslogd's** configuration file, **/etc/syslog.conf**.

Each line in **/etc/syslog.conf** has a “selector” and an “action”. The selector tells which part of the system generated the message and what priority the message has. The action specifies where the message should be sent.

“The part of the selector that tells where a message comes from is called the “facility”. All Internet daemons and servers, except **sendmail**, log messages to the daemon facility. **sendmail** logs messages to the **mail** facility. **syslogd** logs messages to the **syslog** facility. You may indicate all facilities in the configuration file with an asterisk (*****).

The part of the selector that tells what priority a message has is called the “level”. Selector levels are **debug**, **information**, **notice**, **warning**, **error**, **alert**, **emergency**, and **critical**. A message must be at or above the level you specify in order to be logged.

The “action” allows you to specify where messages should be directed. You can have the messages directed to files, users, the console, or to a **syslogd** running on another host.

The following is the default configuration for **/etc/syslog.conf**:

```
mail.debug           /var/adm/syslog/mail.log
*.info,mail.none     /var/adm/syslog/syslog.log
*.alert              /dev/console
*.alert              root
*.emerg              *
```

With this configuration, all mail log messages at the **debug** level or higher are sent to **/var/adm/syslog/mail.log**. Log messages from any facility at the **information** level or higher (but no mail messages) are sent to **/var/adm/syslog/syslog.log**. Log messages from any facility at the **alert** level or higher are sent to the console and any terminal where the superuser is logged in. All messages at the **emergency** level or higher are sent to all users on the system.

Configuring inetd Connection Logging

The **inetd** daemon can log connection requests through **syslogd**. It logs successful connections at the **information** level and unsuccessful connection attempts at the **notice** level. By default, **inetd** starts up with connection logging turned off.

If **inetd** is running with connection logging turned off, issue the following command to start it:

```
# /usr/sbin/inetd -l
```

If **inetd** is running with connection logging turned on, the same command turns it off.

Configuring ftpd Logging

To configure **ftpd** to log messages about logins, login failures, and anonymous **ftp** activity, follow these steps:

1. Add the **-l** or **-v** (verbose) option to the **ftp** line in the **/etc/inetd.conf** file, as in the following example:

   ```
   ftp stream tcp nowait root /usr/lbin/ftpd ftpd -l
   ```

   The **-v** option provides more detailed logging than the **-l** option, except for **anonymous ftp**. For **anonymous ftp**, the **-l** and **-v** options provide the same level of logging.

2. Issue the following command to force **inetd** to read its configuration file:

   ```
   # /usr/sbin/inetd -c
   ```

Configuring Anonymous ftp Access

You can follow the instructions in this section, or you can use SAM to configure **anonymous ftp** access.

1. Use a text editor to add a line for user ftp to the **/etc/passwd** file, as in the following example:

   ```
   ftp:*:500:guest:anonymous ftp:/home/ftp:/usr/bin/false
   ```

   The password field should be *, the group membership should be guest, and the login shell should be **/usr/bin/false**. In this example, user **ftp's** user ID is 500, and the **anonymous ftp** directory is **/home/ftp**.

   Type **man 4 passwd** at the HP-UX prompt for information on the **passwd** file.

2. Create the **ftp** home directory that you configured in the **/etc/passwd** file, as in the following example:

```
# cd /home

# mkdir ftp
```

3. Create the subdirectory **`/usr/bin`** under the **`ftp`** home directory:

```
# cd /home/ftp

# mkdir usr

# cd usr

# mkdir bin
```

4. Copy the **`ls`** and **`pwd`** commands from **`/usr/bin`** to **`~ftp/usr/bin`**, and set the permissions on the commands to 0111 (executable only):

```
# cp /usr/bin/ls /home/ftp/usr/bin

# cp /usr/bin/pwd /home/ftp/usr/bin

# chmod 0111 /home/ftp/usr/bin/ls

# chmod 0111 /home/ftp/usr/bin/pwd
```

5. Set the owner of the **`~ftp/usr/bin`** and **`~ftp/usr`** directories to root, and set the permissions to 0555 (not writeable):

```
# chown root /home/ftp/usr/bin

# chmod 0555 /home/ftp/usr/bin

# chown root /home/ftp/usr

# chmod 0555 /home/ftp/usr
```

6. Create the subdirectory **`etc`** under the **`ftp`** home directory:

```
# cd /home/ftp

# mkdir etc
```

7. Copy **`/etc/passwd`** and **`/etc/group`** to **`~ftp/etc`**. These files are required by the **`ls`** command, to display the owners of files and directories under **`~ftp`**.

```
# cp /etc/passwd /home/ftp/etc

# cp /etc/group /home/ftp/etc
```

8. Replace the password field in all entries in **`/home/ftp/etc/passwd`** with *, and delete the shell field from the end of each entry:

```
ftp:*:500:guest:anonymous ftp:/home/ftp:
acb:*:8996:20::/home/acb:
```

9. Replace the password field in all entries in **`/home/ftp/etc/group`** with *:

```
users:*:20:acb
guest:*:21:ftp
```

10. Set the owner of the files in **`~ftp/etc`** to root, and set the permissions to 0444 (read only):

    ```
    # chown root /home/ftp/etc/passwd

    # chmod 0444 /home/ftp/etc/passwd

    # chown root /home/ftp/etc/group

    # chmod 0444 /home/ftp/etc/group
    ```

11. Set the owner of **`~ftp/etc`** to root, and set the permissions to 0555 (not writeable):

    ```
    # chown root /home/ftp/etc

    # chmod 0555 /home/ftp/etc
    ```

12. Create a directory called **`pub`** and under **`~ftp`**. Set its owner to user **`ftp`** and its permissions to 0777 (writeable by all). **`Anonymous ftp`** users can put files in this directory to make them available to other **`anonymous ftp`** users.

    ```
    # mkdir /home/ftp/pub

    # chown ftp /home/ftp/pub

    # chmod 0777 /home/ftp/pub
    ```

13. Create a directory called **`dist`** and under **`~ftp`**. Set its owner to user root and its permissions to 0755 (writeable only by root). The superuser can put read-only files in this directory to make them available to **`anonymous ftp`** users.

    ```
    # mkdir /home/ftp/dist

    # chown root /home/ftp/dist

    # chmod 0755 /home/ftp/dist
    ```

14. Set the owner of user **`ftp’s`** home directory to root and the permissions to 0555 (not writeable).

    ```
    # chown root /home/ftp

    # chmod 0555 /home/ftp
    ```

Using SAM to Configure Anonymous ftp

1. Start SAM:

    ```
    # sam
    ```

2. Select **`Networking and Communications`**.
3. Select **`Network Services`**.
4. Select **`Anon FTP Deposit`**.
5. Choose **`Actions->Enable`**.

6. Select **`Anon FTP Retrieval`**.
7. Choose **`Actions->Enable`**.

Restricting ftp Access with /etc/ftpusers

When a user attempts to log into your system using **`ftp`**, the **`ftpd`** daemon checks the **`/etc/ftpusers`** file. If the file exists, and the user's login name is listed in it, **`ftpd`** denies access to the user. User accounts that specify a restricted login shell in **`/etc/passwd`** should be listed in **`/etc/ftpusers`**, because **`ftpd`** accesses local accounts without using their login shells. UUCP accounts should also be listed in **`/etc/ftpusers`**.

You can use either a text editor to edit the **`/etc/ftpusers`** file. Each line in **`/etc/ftpusers`** consists of a login name with no white space. Following is an example **`/etc/ftpusers`** file:

```
uucp
guest
nobody
```

For more information, type **`man 4 ftpusers`** at the HP-UX prompt.

Creating /etc/hosts.equiv

If you have already manually configured an **`/etc/hosts.equiv`** file with entries other than those of the form *hostname* or *hostname username* do not use SAM to configure **`/etc/hosts.equiv`**. SAM does not recognize, display or add entries of other forms (such as +, -, %, or +@example_nfsnetgroup).

1. Start SAM

   ```
   # sam
   ```

2. Select the **`Networking and Communications`**.
3. Select the **`System Access.`**
4. Select the **`Remote Logins`**.
5. Choose **`Actions->Add`**.
6. Fill in the form according to its instructions. View the help screens for information about filling in the form.
7. Select **`Apply`** to enter additional names of systems to be configured (use **`Apply`** as a shortcut to remain in the add screen). Then, press **`OK`** when you are done with the screen.

   NOTE: The required choice deletes all **`/etc/hosts.equiv`** file entries for the remote system you specify. Use this choice to remove unwanted entries. You may also modify the list of users that are currently allowed access without a password. Choose **`Select Users Not Required`** and change the list of remote login names as desired.

Summary

TCP/IP is a standard product, but configuring it differs considerably in AIX and HP-UX. Like so many items in AIX, networking configuration by default is part of the ODM configuration database. It is easier to use SMIT to configure TCP/IP because doing so insures that the ODM is brought up to date automatically. You do have the option in AIX to configure TCP/IP using the traditional commands **`ifconfig`** and **`route`**, and to make your changes permanent by either editing the **`/etc/rc.net`** file, in which case you retain SRC support, or editing **`/etc/rc.bsdnet`**, entailing no SRC support.

HP-UX has a traditional means of configuring TCP/IP. However, using SAM can make the process a lot easier. HP-UX also has a **`/var/adm/inetd.sec`** file, which AIX does not, that adds an extra layer of security for TCP/IP.

10. Domain Name Service

AIX

Server Data Files

AIX suggests the following naming convention for **`named`** data files, though you can use whatever scheme suits you:

1. Names and internet addresses of the root name servers: **`named.ca`**
2. Address resolution information for local loopback: **`named.[`*`domain_name`*`]local`**
3. Address resolution data for all machines in the zone: **`named.[`*`domain_name`*`]data`**
4. Reverse address resolution information: **`named.[`*`domain_name`*`]rev`**

These files are usually found in the **`/etc`** directory.

Note: AIX provides sample configuration files in the ***`/usr/samples/tcpip`*** *directory.*

Configuring a Primary Name Server

1. Edit the **`/etc/named.boot`** file, being sure to include the following:
   - Name of the default domain
   - Primary name server designation and names of **`named`** hosts data file and **`named`** reverse hosts data file.
   - Name of the local file (e.g., **`named.local`**)
2. Edit the **`/etc/named.ca`** file to include the names and addresses of the root servers.
3. Edit the **`/etc/named.[`*`domain_name`*`]local`** file. Include the following:
   - Start of authority of the zone and the default time-to-live information.
   - Name server (NS) record.
   - Pointer (PTR) record.
4. Edit the **`/etc/named.[`*`domain_name`*`]data`** file. Include the following:
   - Start of authority of the zone and the default time-to-live information for the zone
   - Name-to-address resolution information on all hosts in the name server's zone of authority
   - Name server records for all primary name servers in the zone
5. Edit the **`/etc/named.[`*`domain_name`*`]rev`** file. Include the following:
   - Start of authority of the zone and the default time to live information

- Address to name resolution information on all hosts to be in the name server's zone of authority

6. Create an empty **`/etc/resolv.conf`** file by issuing the following command:

   ```
   # touch /etc/resolv.conf
   ```

7. Enable the **`named`** daemon using the following SMIT fastpath:

   ```
   # smit stnamed
   ```

   You get the following:

```
                     Start Using the named Subsystem

Move cursor to desired item and press Enter.

  NOW
  Next System RESTART
  BOTH
```

If you choose **`BOTH`**, SMIT starts **`named`** and then edits **`/etc/rc.tcpip`** to get it to start up on each boot.

Configuring a Secondary Name Server

1. Edit the **`/etc/named.boot`** file. This is the same as for a primary server except that you must include secondary lines for each of the domains for which the secondary server is responsible and a secondary line to define the reverse name resolution information. Also, you should include a primary line for the **/etc/named.[*domain_name*]local** file.

2. Edit the **`/etc/named.ca`** file.

3. Edit the **/etc/named.[*domain_name*]local** file.

4. Create an **`/etc/resolv.conf`** file by issuing the following command:

   ```
   # touch /etc/resolv.conf
   ```

   You may want to enter records to specify the name, domain, and address of the name server.

5. Enable the **`named`** daemon using the following SMIT fastpath:

   ```
   # smit stnamed
   ```

Configuring a Cache-Only Name Server

1. Edit the **`/etc/named.boot`** file. Specify a name server type of primary with a source of **`/etc/named.local`** as well as the domain for which the name server will be responsible.

2. Edit the **`/etc/named.ca`** file.

3. Edit the **/etc/named.[*domain_name*]local** file.

4. Create a **`/etc/resolv.conf`** file by issuing the following command:

```
# touch /etc/resolv.conf
```

You may want to enter records to specify the name, domain, and address of the name server.

5. Enable the **named** daemon using the following SMIT fastpath:

   ```
   # smit stnamed
   ```

Setting Up a Remote Name Server

This simply involves creating an **/etc/resolv.conf** file. The easiest way to create, delete, or modify this file is to type:

```
 # smit resolv.conf
```

You get the following:

```
                   Domain Nameserver (/etc/resolv.conf)

Move cursor to desired item and press Enter.

  Start Using the Nameserver
  List All Nameservers
  Add a Nameserver
  Remove a Nameserver
  Stop Using a Nameserver
  -------------------------------
  Set / Show the Domain
  Remove the Domain
  Set / Show the Domain Search List
  Remove the Domain Search List
```

To create a new **/etc/resolv.conf** file, either choose the **Start Using the Nameserver** option, and then the **Create a New /etc/resolv.conf File** option, or you can type the following at the command line:

```
 # smit stnamerslv2
```

See **Setting Up a Remote Name Server** in the HP-UX section for more information on **/etc/resolv.conf**.

Name Resolution Order

If you are using DNS, the resolver attempts to find addresses in the following order:

1. DNS
2. NIS if it running
3. Local **/etc/hosts** if NIS is not running

You can specify a different order by creating the **/etc/netsvc.conf** file, or you can override both the default and **/etc/netsvc.conf** with the `NSORDER` environment variable. **/etc/netsvc.conf** takes the form of:

hosts=*value1,value2,value3*

where ***value*?** Is either **bind**, **nis**, or **local**.

Use the same values in setting the `NSORDER` environment variable, for example:

```
NSORDER=nis,bind,local
```

For more information see **TCP/IP Name Resolution** in Chapter 3 of *System Management Guide: Communications and Networks* in InfoExplorer.

HP-UX

Server Data Files

/etc/named.data/db.cache	Lists the servers for the root domain
/etc/named.data/db.127.0.0.0	Address resolution information for local loopback
/etc/named.data/db.[*domain*]	Address resolution data for all machines in the zone
/etc/named.data/db.[*net*]	Reverse address resolution information

Naming these files **db.[*name*]** is a Hewlett-Packard convention.

Name Server Startup Files

When the system boots to run-level 2 or higher **/sbin/init.d/named** runs and fetches the appropriate variables from the file **/etc/rc.config.d/namesvrs**. The name server starts with the command **/usr/sbin/named**. When invoked, **named** reads the file **/etc/named.boot**, which contains the location of the database files. If you want to start **named** manually, type:

```
 # /sbin/init.d/named start
```

To stop **named** manually:

```
 # /sbin/init.d/named stop
```

Setting the Default Domain Name

If you will be using an **/etc/resolv.conf** file on your host, configure the default domain name with the **search** or **domain** keyword. If you will not be using an **/etc/resolv.conf** file, follow these steps:

1. Set the default domain name with the **hostname** command, by appending the domain name to the host name, as in the following example:

   ```
   # /usr/bin/hostname indigo.div.inc.com
   ```

   Do not put a trailing dot at the end of the domain name.

2. Set the `HOSTNAME` variable in the **/etc/rc.config.d/netconf** file to the same value, as in the following example:

   ```
   HOSTNAME=indigo.div.inc.com
   ```

Configuring a Primary Master Name Server

1. Make sure the **/etc/hosts** file is up to date on the host that will be the primary master server.

2. On the host that will be the primary master, create the **/etc/named.data** directory, where the name server data files will reside, and make it the current directory:

   ```
   # mkdir /etc/named.data
   
   # cd /etc/named.data
   ```

3. Issue the following command to generate the name server data files from the **/etc/hosts** file:

   ```
   # /usr/sbin/hosts_to_named -d domainname -n network_number.
   ```

   Following is an example:

   ```
   # /usr/sbin/hosts_to_named -d div.inc.com -n 15.19.8
   ```

4. Move the **named.boot** file to the **/etc** directory:

   ```
   # mv /etc/named.data/named.boot /etc/named.boot
   ```

5. Copy the file **/usr/examples/bind/db.cache.arpa** to the **/etc/named.data** directory. This file is a list of root name servers. You can also use **anonymous ftp** to get the current list of root name servers from **rs.internic.net**. Instructions are included in the **/usr/examples/bind/db.cache.arpa** file.

6. Use the list of root name servers from the **/usr/examples/bind/db.cache.arpa** file or from **rs.internic.net** to update the **/etc/named.data/db.cache** file. The **hosts_to_named** program creates this file but does not add any data to it.

The **hosts_to_named** program creates the following data files in the directory from which it is run:

- **/etc/named.boot**
- **/etc/named.data/db.cache**
- **/etc/named.data/db.127.0.0**
- **/etc/named.data/db.[*domain*]**
- **/etc/named.data/db.[*net*]**

Naming these files **db.[*name*]** is a Hewlett-Packard convention. You can also create these files manually using a text editor. If you choose to create them manually, you must convert all host names to fully qualified domain names (names containing all labels from the host to the root, terminated with a dot; for example, **indigo.div.inc.com**.)

The **hosts_to_named** program completely rewrites the **db.[*domain*]** and **db.[*net*]** files. All manual modifications to these files will be lost the next time you run **hosts_to_named**, except changes to SOA records. For more information, type **man 1M hosts_to_named** or **man 1M named** at the HP-UX prompt.

To Add a Host to the Domain Data Files

Add the host to **`/etc/hosts`** and run **`hosts_to_named`** again, or add the host manually, as follows:

1. Edit **`db.[`*`domain`*`]`**. Add an Address (A) resource record for each address of the new host. Add CNAME, HINFO, WKS, and MX resource records as necessary. Increment the serial number in the SOA resource record.

2. Edit **`db.[`*`net`*`]`**. Add a PTR resource record for each host address. Increment the serial number in the SOA resource record.

3. Add the host to the **`/etc/hosts`** file. If the host is not listed in **`/etc/hosts`**, someone might run **`hosts_to_named`**, which overwrites your **`db.[`*`domain`*`]`** and **`db.[`*`net`*`]`** files, and the host will be lost.

4. After modifying the domain data files, issue the following command to restart the name server and force it to reload its databases:

```
# /usr/sbin/sig_named restart
```

To Delete a Host from the Domain Data Files

Delete the host from **`/etc/hosts`** and run **`hosts_to_named`** again, or delete the host manually, as follows:

1. Edit **`db.[`*`domain`*`]`**. Delete all A, CNAME, HINFO, WKS, and MX resource records associated with the host. Increment the serial number in the SOA resource record.

2. Edit **`db.[`*`net`*`]`**. Delete all PTR resource records for the host. Increment the serial number in the SOA resource record.

3. After modifying the domain data files, issue the following command to restart the name server and force it to reload its databases:

```
# /usr/sbin/sig_named restart
```

Configuring a Secondary Master Name Server

1. Create a separate directory for the database and configuration files. Choose the same name as on the primary server.

```
# mkdir /etc/named.data

# chmod 755 /etc/named.data
```

2. Copy the **`boot.sec`**, **`boot.sec.save`**, **`db.127.0.0`** files from the primary server.

3. Copy **`db.[`*`net`*`]`** and **`db.[`*`domain`*`]`** if you want local storage.

4. Copy **`boot.sec`** or **`boot.sec.save`** to **`/etc/named.boot`** to create a boot file for **`named`**.

```
# cp /etc/named.data/boot.sec /etc/named.boot
```

5. Copy the **`db.cache`** file from the primary server.

6. Modify **`/etc/namesvrs`** and set the `NAMED` variable to 1.

   ```
   NAMED=1
   ```

7. Create **`/etc/resolv.conf`** to include yourself as a name server.

8. Start **`named`** manually.

   ```
   # /sbin/init.d/named start
   ```

Modifications are made only on the primary server. Secondary servers perform a zone transfer as soon as they discover that the serial number has been incremented. This depends on the configuration of the refresh timers in the SOA records. To initiate an immediate zone transfer on a secondary server restart the server with the **`sig_named`** command.

Configuring a Caching-Only Name Server

The boot file of a caching-only name server has no primary or secondary lines, except the primary line for the **`0.0.127.in-addr.arpa`** domain (the loopback interface). Hosts running Berkeley networking use 127.0.0.1 as the address of the loopback interface. Since the network number 127.0.0 is not assigned to any one site but is used by all hosts running Berkeley networking, each name server must be authoritative for network 127.0.0.

Follow these steps to create a caching-only server:

1. Copy the files **`/etc/named.data/db.127.0.0`** and **`/etc/named.data/db.cache`** from the primary server to the caching-only server.

2. If you ran **`hosts_to_named`** to create the primary master server, **`hosts_to_named`** created a file called **`boot.cacheonly`** in the directory from which it was run. Copy this file to the caching-only server, and rename it **`/etc/named.boot`**.

If you created the primary master server manually, without running **`hosts_to_named`**, create a boot file for the caching-only server called /**`etc/named.boot`**. It should look like the following example:

```
;
; type       domain                        source file
;
directory  /etc/named.data ;running directory for named
primary      0.0.127.IN-ADDR.ARPA          db.127.0.0
cache                                      db.cache
```

Configuring the Resolver to Query a Remote Name Server

Follow these steps if you want your host to query a name server on a remote host:

1. Create a file on your host called **`/etc/resolv.conf`**. The **`/etc/resolv.conf`** file has three configuration options:

   - **`domain`** followed by the default domain name. The domain entry is needed only when the local system's host name (as returned by the hostname command) is not a domain name, and the search option is not configured.

   - **`search`** followed by up to six domains separated by spaces or tabs. The first domain in the search list must be the local domain. The resolver will append these domains, one at a time, to a host name that does not end in a dot, when it constructs queries to send to a name server. The domain and search keywords are mutually exclusive. If you do not specify the **search** option, the default search list will contain only the local domain.

- **nameserver** followed by the internet address (in dot notation) of a name server that the resolver should query.

  You can configure up to three nameserver entries.

The following is an example of **/etc/resolv.conf**:

```
search cs.Berkeley.Edu Berkeley.Edu
nameserver 132.22.0.4
nameserver 132.22.0.12
```

2. If you did not specify the local domain with the **search** or **domain** option, set the default domain name with the **hostname** command, as in the following example,

   ```
   # /usr/bin/hostname indigo.div.inc.com
   ```

   and set the `HOSTNAME` variable in the **/etc/rc.config.d/netconf** file to the same value, as in the following example:

   ```
   HOSTNAME=indigo.div.inc.com
   ```

   Do not put a trailing dot at the end of the domain name.

NOTE: If you want to run both BIND and HP VUE, you must have an **/etc/resolv.conf** file on your system, or HP VUE will not start.

On HP-UX releases before 10.0, by default, if the resolver could not find the requested host by appending the local domain, it would append the parent of the local domain and the grandparent of the local domain. It would not append just the top-level domain (like **com** or **edu**). For example, if BIND could not find host name **aardvark** in the local domain **zoo.bio.nmt.edu**, it would look for **aardvark.bio.nmt.edu** and **aardvark.nmt.edu** but not **aardvark.edu**. On HP-UX release 10.0 and later releases, by default, if you do not specify a search list in **/etc/resolv.conf**, the resolver will append only the local domain to the input host name.

If you want BIND to behave as it did in releases before 10.0, configure a search list in the **/etc/resolv.conf** file. The following search list causes BIND to search the **zoo.bio.nmt.edu** domain as it did by default inreleases before 10.0:

```
search zoo.bio.nmt.edu bio.nmt.edu nmt.edu
```

Starting the Name Server Daemon

Before you start the name server daemon, make sure **syslogd** is running. Follow these steps to start the name server daemon:

1. In the **/etc/rc.config.d/namesvrs** file, set the `NAMED` environment variable to 1, as follows:

   ```
   NAMED=1
   ```

2. Issue the following command to determine whether **named** is already running:

   ```
   # ps -ef | grep named
   ```

If **named** is not running, issue the following command to start it:

```
# /sbin/init.d/named start
```

Configuring the Name Service Switch

The Name Service Switch determines where your system will look for the information that is traditionally stored in the following files:

```
/etc/hosts
/etc/protocols
/etc/services
/etc/networks
/etc/netgroup
/etc/rpc
```

For all types of information except **host** information, you can configure your system to use NIS (one of the NFS Services), the local **/etc** file, or both, in any order. For host information, you can configure your system to use BIND (DNS), NIS, the **/etc/hosts** file, or any combination of the three, in any order. The default Name Service Switch configuration is adequate for most installations, so you probably do not have to change it.

NOTE: Configuring the Name Service Switch is a separate task from configuring the name services themselves. You must also configure the name services before you can use them. The Name Service Switch just determines which name services are queried and in what order. You can use SAM to configure the Name Service Switch.

Following are some suggestions for customizing your Name Service Switch configuration:

- If you want your system to consult the local **/etc/netgroup** file when it fails to find a netgroup in the NIS netgroup database, create or modify the netgroup line in the **/etc/nsswitch.conf** file as follows:

  ```
  netgroup: nis [NOTFOUND=continue] files
  ```

- If you want your system to consult BIND (DNS) when it fails to find a host name in NIS, create or modify the hosts line in the **/etc/nsswitch.conf** file as follows:

  ```
  hosts: nis [NOTFOUND=continue] dns files
  ```

  With this configuration, if NIS does not contain the requested information, and BIND is not configured, the **/etc/hosts** file is consulted.

- If you want your system to consult NIS if it fails to find a host name in BIND or if the BIND name servers are not responding, create or modify the hosts line in the **/etc/nsswitch.conf** file as follows:

- `hosts: dns [NOTFOUND=continue TRYAGAIN=continue] nis files`

With this configuration, if BIND does not return the requested information, and NIS is not running, the **/etc/hosts** file is consulted.

HP recommends that you maintain at least a minimal **/etc/hosts** file that includes important addresses like gateways, diskless boot servers and root servers, and your host’s own IP address.

HP also recommends that you include the word "files" in the hosts line to help ensure a successful system boot using the **`/etc/hosts`** file when BIND and NIS are not available.

CAUTION: Changing the default configuration can complicate troubleshooting. The default configuration is designed to preserve the authority of the name service you are using. It switches from BIND to NIS only if BIND is not enabled. It switches from NIS to the local **`/etc`** file only if NIS is not enabled. It is very difficult to diagnose problems when multiple name servers are configured and enabled for use.

Default Configuration

A default **`nsswitch.conf`** file is supplied in the **`/usr/newconfig/etc`** directory. It contains the following lines :

```
hosts:     dns  nis  files
protocols: nis  files
services:  nis  files
networks:  nis  files
netgroup:  nis  files
rpc:       nis  files
```

This is the default configuration. In other words, if you copy **`/usr/newconfig/etc/nsswitch.conf`** to **`/etc/nsswitch.conf`**, the Name Service Switch behaves the same way it would if no **`/etc/nsswitch.conf`** file existed.

The /etc/nsswitch.conf File

The configuration file for the Name Service Switch is **`/etc/nsswitch.conf`**, which consists of lines with the following syntax:

```
info_type: source [status=action status=action...] source...
```

If the **`/etc/nsswitch.conf`** file does not exist, or if no source is specified in it, the default search order is as follows:

1. DNS (for host information only)
2. NIS
3. Local **`/etc`** file

For more information on the Name Service Switch, type **`man 4 switch`** at the HP-UX prompt.

To Check the Current hosts Configuration

To check the Name Service Switch configuration that your system is currently using for host information, start nslookup and issue the policy command, as follows:

```
# nslookup
> policy
```

The output for the default configuration is as follows:

```
# Lookups = 3
dns [RRCR]     nis [RRCR]     files [RRRR]
```

The letters in square brackets stand for (R)eturn or (C)ontinue. They represent the values of the four status values, `SUCCESS`, `NOTFOUND`, `UNAVAIL`, and `TRYAGAIN`. In the example, the status=action pairs configured for DNS and NIS are

```
SUCCESS=return
NOTFOUND=return
UNAVAIL=continue
TRYAGAIN=return
```

For the following hosts line

```
hosts:  dns [NOTFOUND=continue]  files
```

the **`policy`** command displays the following:

```
# Lookups = 2
dns [RCCR]     files [RRRR]
```

To stop the **`nslookup`** program, type **`exit`**.

To Trace a Host Name Lookup

To trace a host name lookup, start **`nslookup`**, set the **`swtrace`** option, and perform a lookup, as follows:

```
# nslookup
> set swtrace
> hostname
```

For the **`nsswitch.conf`** file containing the hosts line

```
hosts: dns [NOTFOUND=continue] nis [NOTFOUND=continue] files
```

The following example tries all three name services before it finds an answer:

```
# nslookup
> set swtrace
> romney
Name Server: hpindbu.cup.hp.com
Address: 15.13.104.13
lookup source is DNS
Name Server: hpindbu.cup.hp.com
Address: 15.13.104.13
*** hpindbu.cup.hp.com can’t find romney: Non-existent domain
Switching to next source in the policy
lookup source is NIS
Default NIS Server: hpntc43c
Address: 15.13.119.52
Aliases: hpntc43c.cup.hp.com, hpntc43c-119, 3c-119
*** No address information is available for "romney"
Switching to next source in the policy
lookup source is FILES
Using /etc/hosts on: hpntc2k
Name: romney
Address: 15.13.104.128
```

NOTE: If you do not set **`swtrace`**, **`nslookup`** displays only the first name service where it looks for a host, even if it finds the host in another name service.

Updating Network-Related Files

After you configure your system to use BIND, the following network-related configuration files require fully-qualified domain names for all hosts outside your local domain:

```
/etc/hosts.equiv
$HOME/.rhosts
/var/adm/inetd.sec
$HOME/.netrc
```

To Update /etc/hosts.equiv and $HOME/.rhosts

Flat or string-type host names that are not hosts in the local domain must be converted to fully qualified domain names in the **`/etc/hosts.equiv`** file and in all **`$HOME/.rhosts`** files.

The shell script **`convert_rhosts`**, found in **`/usr/examples/bind`**, accepts input conforming to the syntax in **`hosts.equiv`** and converts it to fully qualified domain names. Instructions for using this utility are in the comments at the beginning of the script itself.

To Update /var/adm/inetd.sec and $HOME/.netrc

Flat or string-type host names that are not hosts in the local domain must be converted to fully qualified domain names in the **`/var/adm/inetd.sec`** file and in all **`$HOME/.netrc`** files. No automated utility exists for performing this task, so you must do it manually.

To Update /etc/hosts

To provide an alternate means of lookup if the name server is down, you should maintain a minimal **`/etc/hosts`** file. It should contain the host names and the internet addresses of the hosts in your local domain.

Configuring BIND in SAM

On the local system, you can configure a primary server, a secondary server, a caching-only server, and resolver; start, restart, or stop the server; specify a parent domain, update the DNS database files and configure NS resource records.

More information on configuring BIND in SAM can be found in its help screens. You can get to the DNS section by selecting **`Networking and Communications`** and **`DNS (BIND)`**.

Name Server Statistics

The name server keeps track of various statistics. You can print these statistics to the file **`/var/tmp/named.stats`** by issuing the following command:

```
 # /usr/sbin/sig_named stats
```

Summary

DNS is much the same on both AIX and HP-UX. The differences lie in the names and locations of **`named`** files:

AIX name and location	**HP-UX name and location**
`/etc/resolv.conf`	`/etc/resolv.conf`
`/usr/sbin/named`	`/usr/sbin/named`
`/usr/sbin/named-xfer`	`/usr/sbin/named-xfer`
`/etc/named.boot`	`/etc/named.boot`
`/etc/named.ca`	`/etc/named.data/db.cache`
`/etc/named.[`*domain*`]local`	`/etc/named.data/db.127.0.0`
`/etc/named.[`*domain*`]data`	`/etc/named.data/db.[`*domain*`]`
`/etc/named.[`*domain*`]rev`	`/etc/named.db.[`*net*`]`

11. NIS

AIX

NIS Maps and Commands

Maps and the domainname directory are found in **`/var/yp`**. Files in this directory include:

```
Makefile
aliases.time
binding
group.time
hosts.time
netid.time
passwd.time
protocols.time
publickey.time
rpc.time
rsnsr.hp.com
services.time
updaters
```

The **`make`** command uses the **`Makefile`** in this directory and can be used to create maps and to push them. *Unlike HP-UX, there is no **`ypmake`** script*. **`Makefile`** calls the **`makedbm`** command to build maps, create **`*.time`** files, and push the maps. The **`Makefile`** refers to various **`yp`** commands in the **`/usr/sbin`** directory. The following commands in **`/usr/etc/yp`** are actually links to to commands in **`/usr/sbin`** and are provided for compatibility to previous versions of AIX:

```
chmaster
chslave
chypdom
lsmaster
makedbm
mkalias
mkclient
mkkeyserv
mkmaster
mknetid
mkslave
mrgpwd
revnetgroup
rmkeyserv
rmyp
stdethers
stdhosts
udpublickey
ypinit
yppoll
yppush
ypset
ypxfr
ypxfr_1perday
ypxfr_1perhour
ypxfr_2perday
```

Also, the **yp** commands in **/usr/etc** are actually links:

```
rpc.yppasswdd -> /usr/lib/netsvc/yp/rpc.yppasswdd
rpc.ypupdated -> /usr/lib/netsvc/yp/rpc.ypupdated
ypbind -> /usr/lib/netsvc/yp/ypbind
ypserv -> /usr/lib/netsvc/yp/ypserv
```

Master Server Configuration

You can use SMIT to configure NIS or you can do so manually. If you do a manual configuration, the process is the same as in HP-UX, with the exception of starting the **yp** daemons.

Restricting Access to the Master Server

You can use a password file other than **/etc/passwd**. To do so,

1. Edit the **/etc/rc.nfs** file and change the following stanza:

   ```
   DIR=/etc
   if [ -x /usr/lib/netsvc/yp/rpc.yppasswdd -a -f $DIR/passwd ]; then
           start rpc.yppasswdd /usr/lib/netsvc/yp/rpc.yppasswdd
   /etc/passwd -m
   fi
   ```

2. Change **$DIR/passwd** to the path to your **passwd** file, such as **/etc/passwd.nis**:

   ```
   DIR=/etc
   if [ -x /usr/lib/netsvc/yp/rpc.yppasswdd -a -f $DIR/passwd ]; then
           start rpc.yppasswdd /usr/lib/netsvc/yp/rpc.yppasswdd
   /etc/passwd -m
   fi
   ```

3. Then enter the following command before starting the **yppasswdd** daemon:

   ```
   # chssys -s yppasswdd -a “/etc/passwd.nis -m passwd”
   ```

Creating an NIS Master Server

1. If you haven’t done so already, set the domain name:

   ```
   # smit chypdom
   ```

2. Start SMIT:

   ```
   # smit mkmaster
   ```

3. You will see the following:

```
               Configure this Host as a NIS Master Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
  HOSTS that will be slave servers                  []
* Can existing MAPS for the domain be overwritten?    yes              +
* EXIT on errors, when creating master server?        yes              +
* START the yppasswdd daemon?                         no               +
* START the ypupdated daemon?                         no               +
* START the ypbind daemon?                            yes              +
* START the master server now,                        both             +
   at system restart, or both?
```

4. In the **HOSTS** field, enter the names of your slave servers if you have any or going to have any.

This is all that's required. If you want a more secure setup, you can choose **yes** to **START the yppasswdd daemon** and **START the ypupdated daemon**. See the **man** page for these daemons for more information. If you accept the default value of **both** for **START the master server now, at system restart, or both**, SMIT will call **ypinit -m**, start the appropriate daemons, and make changes to the **/etc/rc.nfs** file to make the changes permanent.

Creating an NIS Slave

1. If you haven't done so already, set the domain name:

   ```
   # smit chypdom
   ```

2. Start SMIT:

   ```
   # smit mkslave
   ```

3. You will see the following:

```
                Configure this Host as a NIS Slave Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* HOSTNAME of the master server                     []
* Can existing MAPS for the domain be overwritten?    yes             +
* START the slave server now,                         both            +
   at system restart, or both?
* Quit if errors are encountered?                     yes             +
```

Enter the name of your master server. If the defaults are fine, press **Enter**. SMIT will run **ypinit -s** and make sure the necessary changes are made to **/etc/rc.nfs** to make the changes permanent.

Creating an NIS Client

1. If you haven't done so already, set the domain name:

   ```
   # smit chypdom
   ```

2. Start SMIT:

   ```
   # smit mkclient
   ```

3. You will see the following:

```
                    Configure this Host as a NIS Client

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* START the NIS client now,                            both            +
   at system restart, or both?
```

Pressing **Enter** at this point starts **ypbind** and edits **/etc/rc.nfs**.

Managing yp Daemons

You can manage all the **yp** daemons using SMIT or on the command line using SRC commands.

1. To use SMIT type:

   ```
   # smit ypstartstop
   ```

2. You will see the following:

```
                    Start / Stop Configured NIS Daemons

Move cursor to desired item and press Enter.

  Start the Server Daemon, ypserv
  Start the Client Daemon, ypbind
  Start the yppasswdd Daemon
  Start the ypupdated Daemon
  Stop the Server Daemon, ypserv
  Stop the Client Daemon, ypbind
  Stop the yppasswdd Daemon
  Stop the ypupdated Daemon
```

3. Follow the menu items to stop or start the appropriate daemons. Each selection above gives you the option of starting or stopping daemons immediately, at system restart, or both.

You can start or stop **yp** daemons using SRC commands, either individually or as a group.

To stop all **yp** daemons:

```
 # stopsrc -g yp
```

To start all **yp** daemons:

```
 # startsrc -g yp
```

To stop a **yp** daemon, for example **ypbind**, individually:

```
 # stopsrc -s ypbind
```

To start a **yp** daemon, for example **ypserv**, individually:

```
 # startsrc -s ypserv
```

Following are excerpts from **/etc/rc.nfs**. To manually configure **yp** in **/etc/rc.nfs**, edit the following stanzas with a text editor:

```
# Uncomment the following lines  and change the domain
# name to define your domain (domain must be defined
# before starting NIS).
if [ -x /usr/bin/domainname ]; then
        /usr/bin/domainname elmo
fi

...

if [ -x /usr/lib/netsvc/yp/ypserv -a -d /var/yp/‘domainname‘ ]; then
      start ypserv /usr/lib/netsvc/yp/ypserv
fi

if [ -x /usr/lib/netsvc/yp/ypbind ]; then
      start ypbind /usr/lib/netsvc/yp/ypbind
fi

if [ -x /usr/sbin/keyserv ]; then
        start keyserv /usr/sbin/keyserv
fi

if [ -x /usr/lib/netsvc/yp/rpc.ypupdated -a -d /var/yp/‘domainname‘ ]; then
        start ypupdated /usr/lib/netsvc/yp/rpc.ypupdated

...

#Uncomment the following lines to start up the NIS
#yppasswd daemon.
DIR=/etc
if [ -x /usr/lib/netsvc/yp/rpc.yppasswdd -a -f $DIR/passwd ]; then
        start rpc.yppasswdd /usr/lib/netsvc/yp/rpc.yppasswdd /etc/passwd -m
fi
```

HP-UX

NIS Maps and Commands

On HP-UX 10.X the following commands are found in **/usr/sbin**:

ypinit
yppoll
yppush
ypset
ypxfr
ypxfrd

The following files as well as the domain directory are found in **/var/yp**:

```
Makefile
binding
securenets
secureservers
updaters
ypmake
ypxfr_1perday
ypxfr_1perhour
ypxfr_2perday
```

The file **`ypmake`** is a shell script that builds one or more maps on a master NIS server. If no arguments are specified, **`ypmake`** either creates maps if they do not already exist or rebuilds maps that are not current. **`yppush`** is then executed to notify slave NIS servers of the change and make the slave servers copy the updated maps to their machines. If any maps are supplied on the command line, **`ypmake`** creates or updates those maps only. The **`make`** command (**`/usr/bin/make`** is a symbolic link to **`/usr/ccs/bin/make`**) can be used instead of **`ypmake.`**

Normal System Startup

All the startup scripts for the NFS and NIS services are started at run level 2 except the **`nfs.server`** script, which is started at run level 3. The following table shows the startup scripts, in the order they are run at system startup. It lists the processes that each script starts and the files and environment variables in **`/etc/rc.config.d`** that influence their behavior.

All of the startup scripts start the portmapper if it is not already started, but only one **`portmap`** process should be running at once.

Startup script in /sbin/init.d	Processes started	Related file in /etc/rc.config.d	Environment variables
nfs.core	portmap	none	none
nis.server	portmap domainname ypserv ypxfrd yppasswdd ypupdated keyserv	namesvrs	NIS_MASTER_SERVER NIS_DOMAIN YPSERV_OPTIONS KEYSERV_OPTIONS YPUDATED_OPTIONS YPXFRD_OPTIONS
nis.client	portmap domainname ypbind keyserv	namesvrs	NIS_CLIENT NIS_DOMAIN WAIT_FOR_NIS_SERVER MAX_NISCHECKS YPBIND_OPTIONS

			KEYSERV_OPTIONS YPSET_ADDR
nfs.client	portmap biod statd lockd automount mount swapon	nfsconf	NFS_CLIENT NUMB_NFSIOD STATD_OPTIONS LOCKD_OPTIONS AUTOMOUNT AUTO_MASTER AUTO_OPTIONS
nfs.server	portmap exportfs mountd nfsd statd lockd pcnfsd swapon	nfsconf	NFS_SERVER NUM_NFSD STATD_OPTIONS LOCKD_OPTIONS START_MOUNTD MOUNTD_OPTIONS PCNFS_SERVER

Configuring and Administering an NIS Master Server

1. Log in as root to the host that will be the master server.
2. On the host that will be the master server, ensure that the PATH environment variable includes the following directory paths:

   ```
   /var/yp
   /usr/lib/netsvc/yp
   /usr/ccs/bin
   ```

3. Issue the following command to set the NIS domain name:

   ```
   # /usr/bin/domainname domainname
   ```

4. In the **`/etc/rc.config.d/namesvrs`** file, set the NIS_DOMAIN variable to the domain name:

   ```
   NIS_DOMAIN=domainname
   ```

5. In the **`/etc/rc.config.d/namesvrs`** file, set the NIS_MASTER_SERVER and NIS_CLIENT variables to 1, as follows:

   ```
   NIS_MASTER_SERVER=1
   ```

```
NIS_CLIENT=1
```

If the host that will be the master server is already a slave server for another domain, set the IS_MASTER_SERVER variable to 1 and the NIS_SLAVE_SERVER variable to 0.

6. Issue the following command to create the NIS maps for the domain:

   ```
   # /usr/sbin/ypinit -m
   ```

   The **ypinit** script will prompt you for the names of your slave servers. Enter the names of your slave servers in response to the prompt.

7. Issue the following commands to run the NIS startup scripts:

   ```
   # /sbin/init.d/nis.server start

   # /sbin/init.d/nis.client start
   ```

   Do not issue the **nis.client start** command until at least one NIS server is running on the same subnet as the client. If you start the NIS client before it has an NIS server to bind to, the client will hang indefinitely.

8. The master server is now running as both an NIS master server and an NIS client. Next, you must configure the slave servers you listed when you ran the **ypinit** script.

Configuring the NIS Master Server to Use a Private passwd File

1. Log in as root to the NIS master server.
2. Copy the **/etc/passwd** file to **/etc/passwd.yp**.
3. Using a text editor, remove users from the **/etc/passwd** file who should not be allowed access to the NIS master server. Do not include a plus sign (**+**) in this file.
4. Use a text editor to edit the **/var/yp/Makefile** file as follows:

   Change the line

   ```
   PWFILE=$(DIR)/passwd
   ```

   to

   ```
   PWFILE=$(DIR)/passwd.yp
   ```

5. In the **/etc/rc.config.d/namesvrs** file, modify the YPPASSWDD_OPTIONS variable as follows:

   Change the line

   ```
   YPPASSWDD_OPTIONS="/etc/passwd -m passwd PWFILE=/etc/passwd"
   ```

   to

   ```
   YPPASSWDD_OPTIONS="/etc/passwd.yp -m passwd PWFILE=/etc/passwd.yp"
   ```

6. Issue the following commands to regenerate the NIS **passwd** maps from **/etc/passwd.yp**:

```
# cd /var/yp
```

```
# /usr/ccs/bin/make passwd
```

This command generates both the **passwd.byname** and the **passwd.byuid** maps and pushes them to the slave servers.

If your slave servers are not up and running yet, run **make** with the NOPUSH flag set to 1:

```
# cd /var/yp
```

```
# /usr/ccs/bin/make NOPUSH=1 passwd
```

This procedure creates a restricted **/etc/passwd** file that is used only by the NIS master server. The unrestricted **/etc/passwd.yp** file is used to generate the NIS passwd maps, which are used by the rest of the hosts in the NIS domain.

Configuring and Administering an NIS Slave Server

The NIS master server must be configured and running before you start your slave servers.

To Edit the Slave Server's passwd File

1. Remove all users from the **/etc/passwd** file except the root user and the system entries required for your system to boot. By convention, system entries usually have user IDs less than 100, so you can remove all entries with user IDs of 100 or greater.

2. Add the following entry as the last line in the **/etc/passwd** file:

   ```
   +::-2:60001:::
   ```

   The plus sign (+) causes processes to consult NIS for any user information not found in the local **/etc/passwd** file. The -2 in the user ID field restricts the access of people who may attempt to log in using "+" as a valid user name when NIS is not running. Anyone who successfully logs in as "+" will be granted only the access missions of user nobody.

   CAUTION: Do not put an asterisk (*) in the password field on HP systems. On Sun systems, an asterisk in the password field prevents people from logging in as "+" when NIS is not running. However, on HP systems, the asterisk prevents all users from logging in when NIS is running.

The changes you make to the **/etc/passwd** file on an NIS slave server are the same changes you make on an NIS client. Following is an example **/etc/passwd** file on an NIS slave server:

```
root:0AnhFBmriKvHA:0:3: :/:/bin/ksh
daemon:*:1:5::/:/bin/sh
bin:*:2:2::/bin:/bin/sh
adm:*:4:4::/usr/adm:/bin/sh
uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico
lp:*:9:7::/usr/spool/lp:/bin/sh
hpdb:*:27:1:ALLBASE:/:/bin/sh
+::-2:60001:::
```

To Enable NIS Slave Server Capability

1. Make sure the NIS master server is already configured and running NIS.
2. Log in as root to the host that will be the slave server.
3. On the host that will be the slave server, ensure that the PATH environment variable includes the following directory paths:

   ```
   /var/yp
   /usr/lib/netsvc/yp
   /usr/ccs/bin
   ```

4. Issue the following command to set the NIS domain name:

   ```
   # /usr/bin/domainname domainname
   ```

   where ***domainname*** is the same as the domain name on the NIS master server.

5. In the **/etc/rc.config.d/namesvrs** file, set the NIS_DOMAIN variable to the domain name:

   ```
   NIS_DOMAIN=domainname
   ```

6. In the **/etc/rc.config.d/namesvrs** file, set the NIS_SLAVE_SERVER and NIS_CLIENT variables to 1, as follows:

   ```
   NIS_SLAVE_SERVER=1
   NIS_CLIENT=1
   ```

   If the slave server is a master server in another NIS domain, set the NIS_MASTER_SERVER variable to 1 and the NIS_SLAVE_SERVER variable to 0. The **yppasswdd** daemon, which is required on the master server, is started only if NIS_MASTER_SERVER=1.

7. Issue the following command to set up the NIS slave server and copy the NIS maps from the master server:

   ```
   # /usr/sbin/ypinit -s NIS_server_name [DOM=domainname]
   ```

   The ***NIS_server_name*** is the name of the master server or a slave server that has a complete set of up-to-date maps for the domain. If the slave server will serve a domain different from the one set by the **domainname** command, specify the domain name after the ***NIS_server_name***.

8. Issue the following commands to run the NIS startup scripts:

   ```
   # /sbin/init.d/nis.server start
   ```

   ```
   # /sbin/init.d/nis.client start
   ```

   Do not issue the **nis.client start** command until at least one NIS server is running on the same subnetwork as the client. If you start the NIS client before it has an NIS server to bind to, the client will hang indefinitely.

In order to receive map updates from the NIS master server, you must add the new slave server to the **ypservers** map on the master server.

To Schedule Regular Map Transfers from the NIS Master Server

1. Log in as root to the slave server.
2. Copy the **`ypxfr_1perday`**, **`ypxfr_2perday`**, and **`ypxfr_1perhour`** scripts from the **`/usr/newconfig/var/yp`** directory to the **`/var/yp`** directory:

   ```
   # cp /usr/newconfig/var/yp/ypxfr_1perday /var/yp
   ```

   ```
   # cp /usr/newconfig/var/yp/ypxfr_2perday /var/yp
   ```

   ```
   # cp /usr/newconfig/var/yp/ypxfr_1perhour /var/yp
   ```

3. Create a **`crontab`** file that invokes these files at regular times. Following is an example **`crontab`** file:

   ```
   0 21 * * * /var/yp/ypxfr_1perday
   30 5,19 * * * /var/yp/ypxfr_2perday
   15 * * * * /var/yp/ypxfr_1perhour
   ```

   This file runs the **`ypxfr_1perday`** script at 9:00 PM every night. It runs the **`ypxfr_2perday`** script at 5:30 AM and 7:30 PM every day. It runs the **`ypxfr_1perhour`** at 15 minutes past every hour.

4. Issue the following command to enter the file into **`crontab`**,

   ```
   # crontab filename
   ```

   where ***`filename`*** is the contab file you just created.

If you have created customized NIS maps for your domain, you will have to add them to the appropriate scripts. You can also use the scripts provided as templates for creating your own scripts.

In some domains, transferring the **`passwd`** maps once per hour generates too much network traffic. If you find this is the case, schedule transfers of the **`passwd`** maps for less frequent intervals.

If you have multiple slave servers, schedule map transfers for different times on different servers, so all the servers are not performing transfers at the same time.

To Add a Slave Server to Your NIS Domain

1. Log in as root to the NIS master server.
2. Issue the following command, where ***`domainname`*** is the name of the domain to which you want to add the slave server:

   ```
   # cd /var/yp/domainname
   ```

3. Issue the following command to create an editable ASCII text file from the ypservers map:

   ```
   # /usr/sbin/makedbm -u ypservers > tempfile
   ```

4. Use a text editor to add the name of the new server to the ASCII file **`tempfile`**.

5. Issue the following command to regenerate the ypservers map from the ASCII file:

   ```
   # /usr/sbin/makedbm tempfile ypservers
   ```

6. Log in as root to the new slave server and configure it as an NIS slave server.

To Remove a Slave Server from Your NIS Domain

1. Log in as root to the NIS master server.
2. Issue the following commands to create an editable ASCII text file from the **`ypservers`** map:

   ```
   # cd /var/yp/domainname
   ```

   ```
   # /usr/sbin/makedbm -u ypservers > tempfile
   ```

3. Use a text editor to remove the name of the slave server from the ASCII file tempfile.
4. Issue the following command to regenerate the **`ypservers`** map from the ASCII file:

   ```
   # /usr/sbin/makedbm tempfile ypservers
   ```

5. Log in as root to the slave server.
6. Remove all the map files from the map directory, and remove the map directory. The directory is called **`/var/yp/`*`domainname`***, where ***`domainname`*** is the name of your NIS domain. For example, if you were removing a slave server from the Finance domain, you would issue the following commands:

   ```
   # cd /var/yp/Finance
   ```

   ```
   # rm *
   ```

   ```
   # cd ..
   ```

   ```
   # rmdir Finance
   ```

7. If the slave is not a slave server in any other NIS domain, use a text editor to set the NIS_SLAVE_SERVER variable to 0 in the **`/etc/rc.config.d/namesvrs`** file.

   ```
   NIS_SLAVE_SERVER=0
   ```

8. If the slave is not a server in any other NIS domain, issue the following command to turn off NIS server capability:

   ```
   # /sbin/init.d/nis.server stop
   ```

To Modify an NIS Map

1. Log in as root to the NIS master server.
2. Make your changes to the source file for the NIS map. For example, if you want to change the NIS hosts map, make your changes to the **`/etc/hosts`** file.
3. Issue the following commands to generate the map and push it to the slave servers:

   ```
   # cd /var/yp
   ```

```
# /usr/ccs/bin/make mapname
```

4. If your slave servers are not up and running yet, run the **make** command with the NOPUSH flag set to 1:

   ```
   # cd /var/yp
   ```

   ```
   # /usr/ccs/bin/make NOPUSH=1 mapname
   ```

   This procedure works for all NIS maps except the **ypservers** map, which has no source file.

5. If you make changes to the **passwd** or **group** maps, regenerate the **netid.byname** map. The **netid.byname** map is a mapping of users to groups, where each user is followed by a list of all the groups to which the user belongs. **The netid.byname** map is generated from the **/etc/passwd** and **/etc/group** files.

Configuring and Administering an NIS Client

To Edit the NIS Client's passwd File

1. Remove all users from the **/etc/passwd** file except the root user and the system entries required for your system to boot. By convention, system entries usually have user IDs less than 100, so you can remove all entries with user IDs of 100 or greater.

2. Add the following entry as the last line in the **/etc/passwd** file:

   ```
   +::-2:60001:::
   ```

   The plus sign (+) causes processes to consult NIS for any user information not found in the local **/etc/passwd** file.

   The -2 in the user ID field restricts the access of people who may attempt to log in using "+" as a valid user name when NIS is not running. Anyone who successfully logs in as "+" will be granted only the access permissions of user nobody.

CAUTION: Do not put an asterisk (*) in the password field on HP systems. On Sun systems, an asterisk in the password field prevents people from logging in as "+" when NIS is not running. However, on HP systems, the asterisk prevents all users from logging in when NIS is running.

The changes you make to the **/etc/passwd** file on an NIS client are the same changes you make on an NIS slave server. Following is an example **/etc/passwd** file on an NIS client:

```
root:0AnhFBmriKvHA:0:3: :/:/bin/ksh
daemon:*:1:5::/:/bin/sh
bin:*:2:2::/bin:/bin/sh
adm:*:4:4::/usr/adm:/bin/sh
uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico
lp:*:9:7::/usr/spool/lp:/bin/sh
hpdb:*:27:1:ALLBASE:/:/bin/sh
+::-2:60001:::
```

To Edit the NIS Client's group File

1. Remove all groups from the **/etc/group** file except the group entries required for your system to boot.

2. Add the following entry as the last line in the **`/etc/group`** file:

   ```
   +:*:*
   ```

   The plus sign (+) causes processes to consult NIS for any group information not found in the local **`/etc/group`** file. The asterisk (*) in the password field prevents people from using the plus sign as a validgroup name if NIS is not running.

The changes you make to the **`/etc/group`** file on an NIS client are the same changes you make on an NIS slave server. Following is an example **`/etc/group`** file on an NIS client:

```
root::0:rootl,sam
other::1:
bin::2:
sys::3:
adm::4:
daemon::5:
mail::6:
lp::7:
+:*:*
```

To Enable NIS Client Capability

1. Make sure at least one NIS master or slave server is running on the client's subnetwork.
2. Log in as root to the NIS client.
3. On the NIS client, ensure that the $PATH environment variable includes the following directory paths:

   ```
   /var/yp
   /usr/lib/netsvc/yp
   /usr/ccs/bin
   ```

4. Issue the following command to set the NIS domain name:

   ```
   # /usr/bin/domainname domainname
   ```

   where domainname is a domain served by an NIS server on the client's subnetwork.
5. In the **`/etc/rc.config.d/namesvrs`** file, set the NIS_DOMAIN variable to the domain name:

   ```
   NIS_DOMAIN=domainname
   ```

6. In the **`/etc/rc.config.d/namesvrs`** file, set the NIS_CLIENT variable to 1, as follows:

   ```
   NIS_CLIENT=1
   ```

7. Issue the following command to run the NIS startup script:

   ```
   # /sbin/init.d/nis.client start
   ```

To Bind an NIS Client to a Server on a Different Subnet

Hewlett-Packard recommends that you configure a server on each subnet where you have NIS clients; however, if you cannot do that, follow these steps to force an NIS client to bind to a server on a different subnet:

1. Log in as root to the NIS client.
2. Add the **`-ypset`** option to the YPBIND_OPTIONS variable in the **`/etc/rc.config.d/namesvrs`** file, as follows:

   ```
   YPBIND_OPTIONS="-ypset"
   ```
3. In the **`/etc/rc.config.d/namesvrs`** file, set the YPSET_ADDR variable to the IP address of an NIS server, as in the following example:

   ```
   YPSET_ADDR="15.13.115.168"
   ```
4. Issue the following commands to restart the NIS client:

   ```
   # /sbin/init.d/nis.client stop
   
   # /sbin/init.d/nis.client start
   ```

If the server you specify in the ypset command is unavailable when your client boots up, your client will broadcast a request for a server to its local network. If no server exists on the local network, the client will hang.

To Query BIND for Host Information After Querying NIS

This section tells you how to set up server-side hostname fallback, which causes your NIS servers to query BIND for host information after querying NIS. A server will search the NIS hosts database first, but if the hosts database does not contain the requested information, the server will query the BIND name service. The server will return the host information to the clients through NIS.

1. Configure your NIS servers as BIND name servers, or install an **`/etc/resolve.conf`** file on each server that allows it to query a BIND name server.
2. On the NIS master server, in the **`/var/yp/Makefile`** file, set the B variable to **`-b`**, as follows:

   ```
   B=-b
   ```
3. Issue the following command on the master server to change the modification time on **`/etc/hosts`** so that make will regenerate the hosts database:

   ```
   # /usr/bin/touch /etc/hosts
   ```
4. Issue the following commands to regenerate the NIS maps on the master server and push them to the NIS slave servers:

   ```
   # cd /var/yp
   
   # /usr/ccs/bin/make
   ```

5. On all the NIS servers in your domain, change the hosts line in the **`/etc/nsswitch.conf`** file to the following:

   ```
   hosts: nis [notfound=continue] dns files
   ```

6. If you do not change the **`/etc/nsswitch.conf`** file, hostname lookups initiated on your NIS servers will be resolved by BIND and not NIS.

CAUTION: Changing the default name service configuration can complicate troubleshooting. The default configuration is designed to preserve the authority of the name service you are using. It switches from BIND to NIS only if BIND is not enabled. It switches from NIS to the **`/etc/hosts`** file only if NIS is not enabled. It is very difficult to diagnose problems when multiple name servers are configured and enabled for use.

Hewlett-Packard recommends that you use the Name Service Switch on your NIS clients instead of server-side hostname fallback. However, if your NIS clients are PCs that do not have a feature like the Name Service Switch, use the server-side hostname fallback described in this section if you want to force BIND lookups after NIS lookups. For more information on the Name Service Switch, see the chapter on DNS in this book.

Summary

There are many common features of NIS on each platform: the **`yp`** daemons have the same name and function, domains work the same, commands like **`ypwhich`**, **`ypset`**, and **`ypinit`** work the same. The differences lie in configuring NIS on each platform. AIX supplies a number of scripts that make configuring NIS relatively easy: **`mkmaster`**, **`mkslave`**, **`mkclient`**, **`chypdom`**. If you use SMIT to configure NIS, then you are actually providing parameters for these scripts. HP-UX does not have equivalent commands, but it does provide the **`ypmake`** script to make map propagation easier. You also use SRC commands in AIX to control the **`yp`** daemons.

Each system has a different way of creating NIS maps: on AIX it involves editing **`/var/yp/Makefile`** while on HP-UX you have to change **`/usr/etc/yp/ypmake`**. Each of these scripts differs in its makeup. However, they ultimately call the **`makedbm`** command and produce maps that can be read across platforms. For example, if you have a map of user account names called **`/etc/auto.user`**, regardless of which platform the corresponding **`.dbm`** and **`.pag`** files are created on, they can be read by NIS utilities on any system, as long as that system is in the same domain.

12. NFS

AIX

Controlling NFS Daemons

You can stop and start most NFS daemons either by using SRC commands or by using SMIT, which ultimately calls SRC commands. The exceptions are **`rpc.rexd`**, **`rpc.ruserd`**, **`rpc.rwalld`**, and **`rpc.rsprayd`**, which are started by **`inetd`**. The following subsystems are part of the **`nfs`** group: **`nfsd`**, **`biod`**, **`rpc.lockd`**, **`rpc.statd`**, and **`rpc.mountd`**.

To start the **`nfs`** group, type:

```
# startsrc -g nfs
```

This starts all of the daemons of the **`nfs`** group as well as the appropriate number of each. For **`nfsd`** and **`biod`** the default number of daemons is eight. To change these defaults, you must do one of the following:

1. Run the **`smit chnfs`** command.
2. Run the **`chnfs`** command, for example:

   ```
   # chnfs -n4 -b4
   ```

3. Run the **`chssys`** command, for example:

   ```
   # chssys -s biod -a6
   ```

The **`chnfs`** command stops currently running **`nfsd`**s and **`biod`**s, updates the ODM database to reflect the new defaults (in this case four daemons each), and then restarts the daemons using SRC commands. The **`chssys`** command changes the **`cmdargs`** descriptor in a subsystem definition, in this case the **`biod`** subsystem. Therefore anytime you run the following command:

```
# startsrc -s biod
```

the value of the **`-a`** parameter is used to determine the number of **`biod`**s to start, in this case six. You can start any subsystem with the **`startsrc -s`** command or stop one with the **`stopsrc -s`** command. For example:

```
# startsrc -s rpc.lockd
# stopsrc -s rpc.mountd
```

The file that controls the NFS (and NIS for that matter) daemons on startup is **`/etc/rc.nfs`**. To disable NFS upon bootup, either remove the line containing that file name from the **`/etc/inittab`** file or run the **`smit rmnfs`** command and select **`restart`** in the **`STOP NFS now, system restart or both`** field. To enable NFS upon bootup it is best to use the **`smit mknfs`** command because it is easy to make a syntax error trying to edit **`/etc/inittab`** by hand.

Note that **`/etc/rc.nfs`** controls the startup of both NFS and NIS. If you want to disable NIS but not NFS you can comment out the appropriate lines in **`/etc/rc.nfs`** or run **`smit rmypserv`** or **`smit rmypclient`**. Do not remove the **`rcnfs`** line in **`/etc/inittab`** if you want only to disable NIS.

Configuring an NFS Server

To configure an NFS server in AIX, all you do is create an **`/etc/exports`** file and then run **`smit mknfs`**. You can use a text editor to create **`/etc/exports`** or you can do the following:

1. Run SMIT:

   ```
   # smit mknfsexp
   ```

2. You will see the following:

```
                     Add a Directory to Exports List

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* PATHNAME of directory to export                   []               /
* MODE to export directory                           read-write      +
  HOSTNAME list. If exported read-mostly            []
  Anonymous UID                                     [-2]
  HOSTS allowed root access                         []
  HOSTS & NETGROUPS allowed client access           []
  Use SECURE option?                                 no              +
* EXPORT directory now, system restart or both       both            +
  PATHNAME of alternate Exports file                []
```

The required entries are **`PATHNAME of directory to export`**, **`MODE to export directory`**, and **`EXPORT directory now, system restart or both`**. Fill in these parameters and press **`Enter`**. SMIT will create or update the **`/etc/exports`** file and then run the **`exportfs -a`** command. The other parameters are optional and can be determined by consulting the **`exports man`** page.

To start the NFS daemons,

1. Run SMIT:

   ```
   # smit mknfs
   ```

2. You will see the following:

```
                                Start NFS

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                       [Entry Fields]
* START NFS now, on system restart or both           both            +
```

Pressing **`Enter`** at this point starts all the daemons of the **`nfs`** group and puts an entry in the **`/etc/inittab`**file to make the changes permanent.

Configuring an NFS Client

You can use the same procedure as above, run **`smit mknfs`** to start the NFS daemons—and in the case of an NFS client the **`biod`**s are the only ones you might want—or you can simply start making NFS mounts from a server by doing the following:

1. Start SMIT:

   ```
   # smit mknfsmnt
   ```

2. You will see the following:

```
                     Add a File System for Mounting

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* PATHNAME of mount point                            []               /
* PATHNAME of remote directory                       []
* HOST where remote directory resides                []
  Mount type NAME                                    []
* Use SECURE mount option?                            no             +
* MOUNT now, add entry to /etc/filesystems or both?   now            +
* /etc/filesystems entry will mount the directory     no             +
   on system RESTART.
* MODE for this NFS file system                       read-write     +
* ATTEMPT mount in foreground or background           background     +
  NUMBER of times to attempt mount                   []               #
Buffer SIZE for read                                 []               #
Buffer SIZE for writes                               []               #
NFS TIMEOUT. In tenths of a second                   []               #
Internet port NUMBER for server                      []               #
* Mount file system soft or hard                      hard           +
  Allow keyboard INTERRUPTS on hard mounts?           yes            +
  Minimum TIME, in seconds, for holding              [3]              #
   attribute cache after f    ile modification
  Maximum TIME, in seconds, for holding              [60]             #
   attribute cache after file modification
  Minimum TIME, in seconds, for holding              [30]             #
   attribute cache after directory modification
  Maximum TIME, in seconds, for holding              [60]             #
   attribute cache after directory modification
  Minimum & Maximum TIME, in seconds, for            []               #
   holding attribute cache after any modification
  The Maximum NUMBER of biod daemons allowed         [6]              #
   to work on this file system
* Allow execution of SUID and sgid programs           yes            +
   in this file system?
* Allow DEVICE access via this mount?                 yes            +
* Server supports long DEVICE NUMBERS?                yes            +
```

The required entries are denoted by the asterisk (**`*`**), are fairly obvious, and have default values which you will in most cases prefer. However, a couple entries need some additional explanation. For **`MOUNT now, add entry to /etc/filesystems or both?`** the default value is **`now`**. If you accept the default, SMIT will make the mount, but the mount will be in effect only until system restart or until you explicitly unmount the file system with the **`umount`** command. Selecting the value of **`both`** will not only result in the mount but also an entry in the **`/etc/filesystems`** file.

As explained in the chapter on disks and file systems, **`/etc/filesystems`** is AIX's equivalent to HP-UX's **`/etc/fstab`** file, which determines which file systems are mounted on bootup. Both files will mount networked file systems as well as local ones; the only difference is their format. If, for example, you mounted the file system **`/doc`** from the machine **`elmo`** and had an entry for that mount in **`/etc/filesystems`**, that entry would look something like:

```
/doc:
        dev             = "/doc"
        vfs             = nfs
        nodename        = elmo
        mount           = false
        options         = bg,hard,intr
        account         = false
```

This example also illustrates the other entry in the **`mknfsmnt`** screen which needs a little explanation: **`/etc/filesystems entry will mount the directory on system RESTART`**. The default value is **`no`**, which results in a **`mount=false`** line in **`/etc/filesystems`**, like in the one above. This means that the remote file system will not be mounted by default upon system restart. If the line reads **`mount=true`**, then the system will attempt to make the NFS mount upon system restart. If it cannot because the server is unavailable, then additional mount requests will occur in the background (**`options=bg`** denotes this). In some cases it is better to have the mount parameter be **`false`**. If, for example, you have several mounts from the same server and it is down, the boot process will not be bogged down by attempting to make remote mounts for each networked file system, in which case it might be better to have an **`/etc/rc.local`** file to complete the mounts once everything else is up and running.

For additional information on the options available in making NFS mounts, see the **`mount`**, **`mknfsmnt`**, and **`filesystems man`** pages.

Additional SMIT Fastpath Commands

Remove an entry from the **`/etc/exports`** file:

```
# smit rmnfsexp
```

Change an exported file system:

```
# smit chnfsexp
```

Remove an NFS mount:

```
# smit rmnfsmnt
```

Configuring Automount

The basic configuration steps are listed below and detailed in the sections that follow.

1. Configure the nodes for NFS if they are not already configured.
2. Create the master map file.
3. Create the map files.

4. If you will be using NIS to administer the maps, integrate the maps with NIS.
5. Start automounter.

Configure for NFS

Configure the systems for NFS if you have not already done so.

Create the Master Map

The master map is usually created as **`/etc/auto.master`** and then made into the NIS map **`auto.master`**. By default, automount tries to get master map information from the NIS map **`auto.master`**. If a local master map file is specified on the command line, automount reads it before reading the NIS **`auto.master map`**.

The format of the **`auto.master`** file is:

```
 DirectoryPath    AutomountMapName
```

where:

- **`DirectoryPath`** is the full pathname of the directory that triggers automount.
- **`AutomountMapName`** is the file that contains the map information.

Create Maps

Automount maps are usually named **`auto.xxx`**, where xxx is the name of the map. The name does not have to correspond to any mount points, but it is recommended that the map name correspond to the directory contents (for example, **`auto.man`** for **`man`** pages).

As with all NIS maps, names must be 10 characters or less if you have file systems that do not allow file names longer than 14 characters. This is because NIS adds four-character suffixes (**`.dir`** and **`.pag`**) to the map name.

The format of the map entries is:

```
 key            [-mount options]             server:directory
```

where:

- ***`key`*** is the name of the subdirectory under the mount directory.
- ***`mount options`*** can be any valid NFS mount options. This field is optional. Mount options specified here override any mount options specified in the master map.
- ***`server:directory`*** specifies the remote server name and the path of the remote file hierarchy (file system or directory) to mount.

Automount recognizes special characters in direct and indirect maps to be used for substitutions and to escape other characters. They are:

&	can substitute key values into the directory path names
*****	is recognized as a catch all entry (a wildcard). It is the last or only entry in a map. It matches all keys and provides a value for the & substitutions that may exist in the right-hand side of a map. For example: `*` -ro,intr server:/users/&
+ mapname	The contents of another map can be included within the current map. If mapname is a directory with no slashes, automount interprets it as an NIS map. If the directory has slashes then automount looks for a local map with that name.

Integrate with NIS

Automount maps can be local files or administered as NIS maps. By default, automount tries to read master map information from the NIS map **`auto.master`**. Automount also reads master map information from a local file if you specify one on the command line.

The master map can contain NIS map names for the indirect and direct maps instead of file names. To specify an NIS map, preface the map name with a plus (+).

To create an NIS map, edit the **`/var/yp/Makefile`**:[9]

1. Add **`auto.master`** to the **`all:`** listing.
2. Add an entry for **`$(DIR)/auto.master`**.
3. Add the following stanza to **`Makefile`**:

```
auto.master.time: $(DIR)/auto.master
   -@if [ -f $(DIR)/auto.master ]; then \
      $(MAKEDBM) $(DIR)/auto.master $(YPDBDIR)/$(DOM)/auto.master; \
      touch auto.master.time; \
      if [ ! $(NOPUSH) ]; then \
         $(YPPUSH) auto.master; \
         echo "pushed auto.master"; \
      else \
         : ; \
      fi \
   else \
      echo "couldn’t find $(DIR)/auto.master"; \
   fi
```

4. Build the map:

```
# cd /var/yp

# make auto.master
```

[9]Taken from *NIS Automounter, Infoexplorer*

Start Automounter

The full pathname in AIX is **`/usr/sbin/automount.`** The options found in the **`automount man`** page are the same as those for the HP-UX version. You can also start automount by typing the **`smit mkautomnt`** command. If you want to start automount on bootup, the best place to put it is at the end of the **`/etc/rc.nfs`** file, something you will have to do manually since SMIT doesn't do it for you.

Shutting Down Automount

`automount` is normally started and stopped only when the machine is rebooted. To shut down **`automount`** gracefully during system operation, take the following steps:

1. Make sure no processes have their current working directory set to any **`automount`** directories or subdirectories.
2. Then, send **`automount`** the SIGTERM (-15) signal (SIGTERM is the default signal sent by the **`kill`** command).

CAUTION: No other **`automount`** daemon should be started until the first has successfully cleaned up and exited. If a second **`automount`** daemon is started when the first is in its shutdown process, the second daemon will start its shutdown process. This means that there will now be four **`automount`** daemons: the first, the second, and their children. These daemons will not exit until all the mount directories they are serving have been unmounted.

Do not send the SIGKILL signal (**`kill -9`**, **`kill -KILL`**) to the **`automount`** daemon. This will cause any processes accessing mount directories served by **`automount`** to hang. The file hierarchies mounted by **`automount`** under **`/tmp_mnt`** will still be accessible

HP-UX

Installing the NFS Services Software

NFS is typically bundled with the HP-UX operating system. The NFS product was installed when you installed your operating system and is also installed on preconfigured systems. If you do not have the NFS product, you can install it from separate media or from a network server using the **`swinstall`** command.

Startup Files

The startup scripts for core and client NFS functionality are **`/sbin/init.d/nfs.core`** and **`/sbin/init.d/nfs.client`**. These are run at run-level 2. At run-level 3 **`/sbin/init.d/nfs.server`** enables NFS server functionality. The run scripts are configured by the **`/etc/rc.config.d/nfsconf`** file.

Configuring and Administering an NFS Server

Exporting directories

1. Add a line to the **`/etc/exports`** file for each directory you want to make available to NFS clients, using a text editor like vi. If the **`/etc/exports`** file does not exist on your system, you will have to create it. Following is the syntax of a line in the **`/etc/exports`** file:

```
directory [-option[,option]]
```

Type man 4 exports at the HP-UX prompt for a complete list of the export options. After adding your exported directories to the **/etc/exports** file, you must enable NFS server capability before NFS clients can mount your exported directories

2. If your system is already running as an NFS server, issue the following command to add the directory to your server's internal list of exported directories:

```
# /usr/sbin/exportfs directory
```

You can issue the **exportfs** command without adding the directory to the **/etc/exports** file, and the directory will be added to your server's internal list of exported directories. However, it will stop being exported when you reboot your system or restart NFS, unless you also add it to the **/etc/exports** file. (Issuing the **exportfs** command does not change the contents of the **/etc/exports** file.) Type man 1M exportfs for more information. The **/etc/exports** file should be owned by root and have mode 644 (-rw-r--r--).

To Enable NFS Server Capability

1. In the **/etc/rc.config.d/nfsconf** file, make sure the NFS_SERVER and START_MOUNTD variables are set to 1, as follows:

```
NFS_SERVER=1
START_MOUNTD=1
```

2. Issue the following command to run the NFS startup script:

```
# /sbin/init.d/nfs.server start
```

The NFS startup script uses the variables in **/etc/rc.config.d/nfsconf** to determine which processes to start. The START_MOUNTD variable causes the NFS startup script to start **rpc.mountd**, the **mount** daemon.

If you need strict security, set the START_MOUNTD variable to 0, and configure the **rpc.mountd** daemon with the **-e** option in the **/etc/inetd.conf** file. This causes **inetd** to restart **rpc.mountd** with each mount request and to check credentials and access permissions for each request.

When **rpc.mountd** is started with the **-e** option, no TCP port is opened for it. This can cause problems if your NFS clients use the automounter **-hosts** map, and one of the hosts on your network has a very large **/etc/exports** file. Before an NFS client automounts directories from a server, it polls the server for a list of its exported directories. If the list is very long, and no TCP port is open for **rpc.mountd**, the list may become truncated when it is sent over UDP, and the client will be able to **automount** only a subset of the exported directories from that host.

CAUTION: **If rpc.mountd** is configured in **/etc/inetd.conf** on your system, set the START_MOUNTD flag to 0. Mounts will fail if **rpc.mountd** is enabled through both **/etc/inetd.conf** and **/etc/rc.config.d/nfsconf**.

Unexporting Directories

1. On the NFS server, issue the following command for a list of all the NFS clients that have mounted the directory you want to unexport:

```
# /usr/sbin/showmount -a
```

2. On every NFS client that has the directory mounted, issue the following command for a list of the process IDs and user names of everyone using the mounted directory:

   ```
   # /usr/sbin/fuser -u servername:/directory
   ```

3. Warn any users to cd out of the directory, and kill any processes that are using the directory, or wait 3 until the processes terminate. You can use the following command to kill all processes using the directory:

   ```
   # /usr/sbin/fuser -ck local_mount_point
   ```

4. On every NFS client that has the directory mounted, issue the following command to unmount the directory:

   ```
   # /usr/sbin/umount servername:/directory
   ```

5. On every NFS client that had the directory mounted, use a text editor to comment out or remove the line in the **/etc/fstab** file that lists the directory you want to unexport. This prevents clients from attempting to mount the directory when they reboot.

6. On every client that has the directory configured to be automounted, edit the **/etc/auto_*** files to comment out or remove the directory from the automounter maps. Clients that **automount** the directory may not be listed by the **showmount** command.

   If you are using NIS to manage your automounter maps, edit the **/etc/auto_*** files on the NIS master server, and then issue the following commands to regenerate the maps and push them to the slave servers:

   ```
   # cd /var/yp
   
   # /usr/ccs/bin/make auto.mapname auto.mapname...
   ```

7. If you modified any direct automounter maps or the automounter master map, restart the automounter.

8. On the NFS server, use a text editor to remove the line in the **/etc/exports** file that lists the directory you want to unexport.

9. On the NFS server, issue the following command to unexport the directory:

   ```
   # /usr/sbin/exportfs -u directory
   ```

If you unexport a directory that an NFS client currently has mounted, the next time someone on that client requests access to the directory, NFS will return an NFS stale file handle error message. The client may be able to unmount the directory, but if that does not work, the client must reboot to recover.

To Disable NFS Server Capability

1. On the NFS server, issue the following command for a list of all the NFS clients that have directories mounted from the NFS server you are planning to disable:

   ```
   # /usr/sbin/showmount -a
   ```

2. On every NFS client listed by the showmount command, issue the following command for each directory that is mounted from your NFS server:

   ```
   # /usr/sbin/fuser -u servername:/directory
   ```

   This command lists the process IDs and user names of everyone using the mounted directory.

3. Warn any users to cd out of the directory, and kill any processes that are using the directory, or wait until the processes terminate. You can use the following command to kill all processes using the directory:

   ```
   # /usr/sbin/fuser -ck local_mount_point
   ```

4. On every client that has directories mounted from your server, issue the following command:

   ```
   # /usr/sbin/umount -h servername
   ```

5. If your server will be down for a long time, edit the **/etc/fstab** file on each client to comment out or remove any NFS mounts from the server you are planning to disable. This prevents the clients from attempting to mount directories from your server when the clients are rebooted.

6. If your server will be down for a long time, edit the **/etc/auto_*** files on each client to comment out or remove any automounts from the server you are planning to disable. Clients that automount the server's directories might not be listed by the **showmount** command.

   If you are using NIS to manage your automounter maps, edit the **/etc/auto_*** files on the NIS master server, and then issue the following commands to regenerate the maps and push them to the slave servers:

   ```
   # cd /var/yp

   # /usr/ccs/bin/make auto.mapname auto.mapname...
   ```

7. If you modified any direct automounter maps or the automounter master map, restart the automounter.

8. Issue the following command on the server to unexport all exported directories:

   ```
   # /usr/sbin/exportfs -au
   ```

9. On the NFS server, edit the **/etc/rc.config.d/nfsconf** file to set the NFS_SERVER variable to 0.This prevents the NFS server daemons from starting up when your system reboots. If your server will be down only a short time, this step is unnecessary.

   ```
   NFS_SERVER=0
   ```

10. Edit the **/etc/inetd.conf** file to comment out the line that contains **rpc.mountd** (if it exists) and the lines for the other RPC services.

11. Issue the following command to disable NFS server capability:

    ```
    # /sbin/init.d/nfs.server stop
    ```

If your NFS server will be down for only a very short period of time, this procedure is not necessary. If the server is down for only a few minutes, and directories are hard-mounted on the clients, clients attempting access to the server will simply hang until it comes back up. Then, they will resume access to it as if nothing had happened.

However, if the server will be down for a long time, NFS clients attempting access to it will have to interrupt their attempts, usually with [CTRL]-C. If directories are mounted with the **`nointr`** option, clients must reboot their systems in order to stop trying to access a down server.

Configuring and Administering an NFS Client

To Mount a Remote Directory Using a Standard NFS Mount

1. In the **`/etc/fstab`** file, use a text editor to add a line for each remote directory you want mounted on your system. If the **`/etc/fstab`** file does not exist, you will have to create it. A line in the **`/etc/fstab`** file has the following syntax:

   ```
   server:remote_directory local_directory nfs defaults 0 0
   server:remote_directory local_directory nfs option[,option...] 0 0
   ```

2. If your system is already running as an NFS client, issue the following command to mount each remote directory you have added to the **`/etc/fstab`** file:

   ```
   # /usr/sbin/mount local_directory
   ```

   Or, issue the following command to mount all the directories listed in the **`/etc/fstab`** file:

   ```
   # /usr/sbin/mount -a
   ```

The remote directories listed in the **`/etc/fstab`** file will be mounted automatically when you enable NFS client capability or reboot your system. Before you can mount a remote directory on your system, the remote system where the directory is located must be configured as an NFS server and must export the directory.

To mount a directory temporarily, issue the **`mount`** command, but do not add the mount to the **`/etc/fstab`** file. It will stay mounted until you reboot your system or until you unmount it with the umount command.

To Enable NFS Client Capability

1. In the **`/etc/rc.config.d/nfsconf`** file, make sure the NFS_CLIENT variable is set to 1, as follows:

   ```
   NFS_CLIENT=1
   ```

2. Run the NFS startup script by issuing the following command:

   ```
   # /sbin/init.d/nfs.client start
   ```

Setting the NFS_CLIENT variable to 1 causes the NFS startup script to be run whenever you reboot your system. The NFS startup script starts the necessary NFS client daemons and mounts the remote directories configured in the **`/etc/fstab`** file.

To Disable NFS Client Capability

1. On the NFS client, issue the **`mount`** command with no options, to get a list of all the mounted file systems on the client:

   ```
   # /usr/sbin/mount
   ```

2. For every NFS-mounted directory listed by the mount command, issue the following command to determine whether the directory is currently in use:

   ```
   # /usr/sbin/fuser -cu local_mount_point
   ```

   This command lists the process IDs and user names of everyone using the mounted directory.

3. Warn any users to cd out of the directory, and kill any processes that are using the directory, or wait until the processes terminate. You can use the following command to kill all processes using the mounted directory:

   ```
   # /usr/sbin/fuser -ck local_mount_point
   ```

4. Issue the following command on the client to unmount all NFS-mounted directories:

   ```
   # /usr/sbin/umount -at nfs
   ```

5. Edit the **`/etc/rc.config.d/nfsconf`** file on the client to set the NFS_CLIENT and AUTOMOUNT variables to 0. This prevents the client processes from starting up again when you reboot the client.

   ```
   NFS_CLIENT=0
   AUTOMOUNT=0
   ```

6. Issue the following command to disable NFS client capability:

   ```
   # /sbin/init.d/nfs.client stop
   ```

Automounter

The master map is usually created as **`/etc/auto_master`** and then made into the NIS map **`auto_master`**. By default automounter tries to get master information from the the **`auto_master`** map. If a local master map file is specified on the command line, automounter reads it before reading **`auto_master`**.

The following is the format for indirect map entries:

mount_directory   *map_name*   *[mount_options]*

The following is the format for direct map entries

/-   *direct_map*   *[mount_options]*

in which

`mount_directory` is the absolute path of the mount directory for the indirect map, ***`/-`*** is the direct map indicator, and ***`map_name/direct_map`*** is the file name or NIS map name of the indirect or direct map. If the map name is prefaced with a plus (+), automounter searches for an NIS map.

The ***`map_name`*** or ***`direct_map`*** field can also reference on of the following special entries:

`-hosts`	A special map that uses **`/etc/hosts`**, NIS, or BIND to locate a remote when a host name is specified as a directory name. The name of the remote host is used as the subdirectory name under the mount directory (the first field in the master map)
`-password`	A special map that uses the **`/etc/passwd`** file to attempt to locate the home directory of a user.
`-null`	Cancels a previous map for the indicated mount directory.

`mount_options` refers to any NFS mount options. Mount options in the master map are overridden by mount options in the indirect/direct maps.

Automounter Startup Files

When booted to run-level 2 or higher, the system will run the **`/sbini/init.d/nfs.client`** script. This script will start automounter if the AUTOMOUNT variable in the **`/etc/rc.config.d/nfsconf`** configuration file has a value of 1.

Configuring Automounter

1. Create the master map **`/etc/auto_master`**.
2. Create indirect and direct maps as needed.
3. Configure **`/etc/rc.config.d/nfsconf`**.
4. Start automounter manually:

   ```
   # /sbin/init.d/nfs.client start.
   ```

To Enable the NFS Automounter

1. In the **`/etc/rc.config.d/nfsconf`** file, make sure the NFS_CLIENT and AUTOMOUNT variables are set to 1, as follows:

   ```
   NFS_CLIENT=1
   AUTOMOUNT=1
   ```

2. Make sure the AUTO_MASTER variable in **`/etc/rc.config.d/nfsconf`** is set to the name of your automounter master map. (The default master map name is **`/etc/auto_master`**.)

   ```
   AUTO_MASTER="/etc/auto_master"
   ```

3. Issue the following command to run the NFS client startup script:

   ```
   # /sbin/init.d/nfs.client start
   ```

To Restart the Automounter

1. Issue the following command to get a list of all the automounted directories on the client:

   ```
   # /usr/bin/grep tmp_mnt /etc/mnttab
   ```

2. For every automounted directory listed by the **`grep`** command, issue the following command to determine whether the directory is currently in use:

   ```
   # /usr/sbin/fuser -cu local_mount_point
   ```

3. Warn any users to **`cd`** out of the directory, and kill any processes that are using the directory, or wait until the processes terminate. You can issue the following command to kill all the processes using the mounted directory:

   ```
   # /usr/sbin/fuser -ck local_mount_point
   ```

4. Issue the following commands to kill the automounter (PID is the process ID returned by the **`ps`** command):

   ```
   # ps -ef | grep automount
   ```

   ```
   # kill -SIGTERM PID
   ```

   CAUTION: Do not kill the automounter with -SIGKILL (-9). The SIGKILL signal can cause any currently automounted directories to become inaccessible until you reboot your system.

5. Type the **`ps`** command to make sure the automounter is no longer active:

   ```
   # /usr/bin/ps -ef | grep automount
   ```

   If the **`ps`** command indicates the automounter is still active, make sure all users are out of the automounted directories and then try again. Do not restart the automounter until all automount processes have terminated.

6. Issue the following command to start the automounter:

   ```
   # /usr/sbin/automount [options]
   ```

   `options` is the list of options configured in the AUTO_OPTIONS variable in the **`/etc/rc.config.d/nfsconf`** file. You can also source the **`/etc/rc.config.d/nfsconf`** file, and then enter the automount command as follows:

   ```
   # /usr/sbin/automount $AUTO_OPTIONS
   ```

Summary

NFS, originally a Sun product, works mostly the same in both AIX and HP-UX. However, some of the details differ, as always. AIX controls NFS daemons through the System Resource Controller (SRC) while HP-UX allows for direct commands. Both operating systems' system management tools (SMIT and SAM) support NFS configuration, which in most cases is the preferred way to go. Automatic NFS mounts on bootup are controlled by **`/etc/filesystems`** (AIX) and **`/etc/fstab`** (HP-UX). Though the format of each file differs, the principle in each is the same.

Both systems support automount, though you have to configure it manually on each. The biggest difference in the automount area is its integration with NIS. Otherwise, NFS has the same "look and feel" across AIX and HP-UX platforms. Both platforms export file systems listed in **`/etc/exports`** via the **`exportfs`** command. Both systems have **`nfsd`**s, **`biod`**s**`, rpc.mountd`**, **`rpc.lockd`**, and **`rpc.statd`**. Since NFS uses the External Data Representation (XDR) protocol, there is no problem in data representation across the two systems.

13. Mail

AIX

AIX user agents are **`mail`**, **`mailx`**, and **`Mail`**, all of which are the same program since these are all hard links to the same executable. This program most closely resembles the **`mailx`** program found on HP-UX systems. Another user agent, **`mhmail`**, just composes, sends, and files messages. The default routing agent is the **`sendmail`** program, and the default delivery agents are **`/usr/bin/bellmail`** (for local users), Basic Networking Utilities (BNU, a version of UUCP), and SMTP (for TCP/IP).

Customizing the /etc/sendmail.cf file

The default **`/etc/sendmail.cf`** file defines configuration for local, TCP/IP, and BNU delivery. This should be enough for most sites, but if you need further customization, do the following:

1. Edit the **`/etc/sendmail.cf`** file. You can use the **`vi`** editor or an editor that does not convert tabs to characters, or you can use the **`/usr/lib/edconfig`** command.
2. Create the database file, **`/etc/sendmail.cfDB`** with the following command:

   ```
   # sendmail -bz
   ```

3. Refresh the **`sendmail`** daemon:

   ```
   # refresh -s sendmail
   ```

Stopping and Starting the sendmail Daemon

`sendmail` by default should start up at bootup. If it doesn't, then uncomment the following line from **`/etc/rc.tcpip`**:

```
 start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"
```

To start the **`sendmail`** daemon using the SRC:

```
 # startsrc -s sendmail -a "-bd -q30m"
```

To stop the **`sendmail`** daemon using the SRC type the following:

```
 # stopsrc -s sendmail
```

To start the **`sendmail`** daemon using the command line:

```
 # sendmail -bd -q30m
```

To stop the **`sendmail`** daemon using the Korn shell **`kill`** command:

```
 # kill $(cat /etc/sendmail.pid)
```

Mail Aliases

AIX keeps mail aliases in **`/etc/aliases`**. In addition to aliases you provide, this file must contain an alias for each of the following:

- MAILER-DAEMON (usually set to root)
- postmaster (usually set to root)
- nobody (usually set to **`/dev/null`**)

To compile the aliases database file type one of the following:

```
# sendmail -bi
# newaliases
```

This command creates the following dbm files:

- **`/etc/aliasesDB/DB.dir`**
- **`/etc/aliasesDB/DB.pag`**

The Mail Queue

In AIX the mail queue directory is **`/var/spool/mqueue`**. The naming convention for the files in this directory is virtually the same as that for HP-UX:

A file beginning with	**is a**
df	data file
qf	queue control file
tf	temporary file
xf	transcript file
nf	backup file
lf	lock file

The **`sendmail`** queue processing interval, as in HP-UX, is set with the **`-q`** option to the **`sendmail`** command.

To process the mail queue once:

```
# sendmail -q
```

To process the mail queue every 15 minutes:

```
# sendmail -q15m
```

To process the mail queue every 30 seconds:

```
# sendmail -q30s
```

To change the processing interval permanently, edit the following line in **`/etc/rc.tcpip`** to the value you desire:

```
qpi=30m
```

The **`qpi`** variable is used as an argument in the next line:

```
start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"
```

Mail Logging and Statistics

Mail logging by default goes to the file **`/var/spool/mqueue/log`**. This is specified in the **`/etc/syslog.conf`** file. If you want to log to a different file just edit **`/etc/syslog.conf`** and refresh **`syslogd`**:

```
# refresh -s syslogd
```

AIX supports the **`mailstats`** command. For this command to work the **`/etc/sendmail.st`** file must exist with permissions 660.

Sending AIX Mail to an HP-UX Hub

Configuring AIX's **`sendmail.cf`** file to route to an HP-UX hub is quite simple. Consider the following lines from **`/etc/sendmail.cf`**:

```
###################################################################
#
# Host name for central mail server  (YOU MAY OPTIONALLY DEFINE THIS)
#
# Optionally defined macro specifying name of host to which you want
# to relay all local mail.  This is used if there is a central mail
# server that should receive all mail for a given area.
#
# When this macro is defined, mail that would otherwise be delivered
# to "user" on the local machine is delivered to "user@server" on the
# server machine.  Note that this is similar to what would occur if
# the mail spool directory were remote-mounted from the mail server.
# For this reason, this macro can be used to achieve the same effect
# as remote-mounting the mail spool directory, but without the access
# and locking problems that can sometimes occur with remotely mounted
# filesystems.
#
#DLMailServerName
#
###################################################################
```

If, for example, your mail hub's hostname was **`elmo`**, you would need to create a line that reads:

```
DLelmo
```

Mail sent to **`dylan@sadie`**, for example, would be directed to the machine **`elmo`**, even if **`elmo`** was an HP-UX box, provided of course that **`sadie`**'s **`/etc/sendmail.cf`** file contains the above line and that the user **`dylan`** was recognized on both systems. The user **`dylan`** would then have to either log on to **`sadie`** to get his messages, or the **`/usr/mail`** directory on **`sadie`** would have to be mounted on **`/var/spool/mail`** on **`elmo`**. NFS mounts of the respective mail spooling directories on both systems seem to have no problem with permissions as long as they are exported read-write with either root access granted to machines within the domain or with the **`anonymous`** userid set to 0.

Remember that making the above change in **`/etc/sendmail.cf`** is not enough: you have to rebuild **`/etc/sendmail.cfDB`** and refresh the **`sendmail`** daemon.

HP-UX

When you install **`sendmail`**, the installation script creates and modifies files on the system that are needed for **`sendmail`** operation. The **`sendmail`** configuration file supplied with HP-UX

10.20 will work without modifications for most installations. Therefore, the only steps you must do are: set up **`sendmail`** servers to run with NFS, configure and start **`sendmail`** clients, and verify that **`sendmail`** is running properly.

Installing sendmail on a Standalone System

When **`sendmail`** is installed, it is automatically configured to send and receive mail for users on the local system only. The standalone system processes all outbound mail and establishes connections to the message destination host or to Mail Exchanger (MX) hosts. The **`sendmail`** daemon is then started when you reboot the system, so you do not need to make any changes to any system files.

Installing sendmail on a Mail Server

To set the system up as an NFS server and allow the **`sendmail`** clients to read and write to the **`/var/mail`** directory, do the following:

1. Make sure all mail users have accounts on the mail server and that their user IDs and group IDs on the mail server are the same as on the client machines. (This step is not necessary if you are using NIS and your mail server is in the same NIS domain as the clients.)
2. In the **`/etc/rc.config.d/nfsconf`** file, use a text editor to set the NFS_SERVER variable to 1.
3. Use a text editor to add the following line to the **`/etc/exports`** file:

   ```
   /var/mail  -access=client,client...
   ```

   where each mail client is listed in the access list. If the **`/etc/exports`** file does not exist, you will have to create it.
4. Issue the following command to run the NFS startup script:

   ```
   # /sbin/init.d/nfs.server  start
   ```

Installing sendmail on a Mail Client

`sendmail` clients do not receive mail on their local system; instead, users on the client systems obtain their mail on the mail server. User mail directories reside on the server, and users read their mail over an NFS link. By default, a **`sendmail`** client forwards to the server any local mail (a user address destined for the client system) and sends non-local mail directly to the destination system or MX host. Outgoing mail appears to originate from the server, so replies are sent to the server. **`sendmail`** clients can be diskless systems.

To configure a sendmail client system to access a sendmail server:

1. In the **`/etc/rc.config.d/mailservs`** file, use a text editor to set the SENDMAIL_SERVER variable to 0. This ensures that the sendmail daemon will not be started when you reboot your system or run the sendmail startup script.
2. In the **`/etc/rc.config.d/mailservs`** file, use a text editor to set the SENDMAIL_SERVER_NAME variable to the host name or IP address of the mail server you will use (the machine that will run the sendmail daemon).

3. In the **`/etc/rc.config.d/nfsconf`** file, use a text editor to set the NFS_CLIENT variable to 1.
4. Use a text editor to add the following line to the **`/etc/fstab`** file:

   ```
   servername:/var/mail  /var/mail  nfs  0  0
   ```

   where servername is the name configured in the SENDMAIL_SERVER_NAME variable in **`/etc/rc.config.d/mailservs`**. If the **`/etc/fstab`** file does not exist, you will have to create it.
5. Issue the following command to run the **`sendmail`** startup script:

   ```
   # /sbin/init.d/sendmail start
   ```
6. Issue the following command to run the NFS startup script:

   ```
   # /sbin/init.d/nfs.client start
   ```

The **`sendmail`** startup script assumes that this system will use the host specified by the SENDMAIL_SERVER_NAME variable as the mail hub. The script also assumes that mail sent from this system should appear to be from the host specified by the SENDMAIL_SERVER_NAME variable (this feature may previously have been known as "site hiding"). The script therefore modifies the macros DM (for "masquerade") and DH (for "mail hub") in the system's **`/etc/mail/sendmail.cf`** file to use the host specified by the SENDMAIL_SERVER_NAME variable. Note that if the DM and DH macros have previously been defined, the startup script does not modify them. As mentioned earlier, the client system now forwards local mail to the mail server and forwards other mail directly to remote systems.

To configure the client system to relay all mail to the mail server for delivery, see "Modifying the Default sendmail Configuration File" . The NFS startup script NFS-mounts the /var/mail directory from the mail server to your system.

Adding sendmail Aliases to the Alias Database

1. If the file **`/etc/mail/aliases`** does not exist on your system, copy it from **`/usr/newconfig/etc/mail/aliases`** to **`/etc/mail/aliases`**.
2. Use a text editor to add lines to the file. Each line has the following form:

   ```
   alias : mailing_list
   ```

   where **`alias`** is a local address, local user name, or local alias, and **`mailing_list`** is a comma-separated list of local user names or aliases, remote addresses, file names, commands, or included files.
3. Issue the following command to regenerate the aliases database from the **`/etc/mail/aliases`** file:

   ```
   # /usr/sbin/newaliases
   ```

   This command creates the aliases database, which is located in the file **`/etc/mail/aliases.db`**.

Restarting sendmail

Issue the following commands, on a standalone system or on the mail server, to restart sendmail:

```
# /sbin/init.d/sendmail stop
# /sbin/init.d/sendmail start
```

Configuring a sendmail Client to Relay All Mail to a Server.

As mentioned previously, the default behavior for a **`sendmail`** client is to forward only local mail to the **`sendmail`** server for delivery; the client system sends non-local mail directly to remote systems. If you want the client to relay all mail (local and non-local) to the **`sendmail`** server, you will need to build a new **`sendmail.cf`** file using the **`m4`** macros. (**`m4`** macros are a set of library routines used to create customized **`sendmail`** configuration files.) Follow these steps to create a new **`sendmail.cf`** file:

1. Change directory to the **`/usr/newconfig/etc/mail/cf/cf`** directory:

   ```
   # cd /usr/newconfig/etc/mail/cf/cf
   ```

2. Copy the file **`examples/clientproto.mc`** to this directory:

   ```
   # cp examples/clientproto.mc.
   ```

3. Edit the following statements in the **`clientproto.mc`** file:

   ```
   OSTYPE(hpux10)

   ...

   FEATURE(mailhost.$m)
   ```

   For **`mailhost`**, enter the hostname of the **`sendmail`** server.

4. Use the **`m4`** macros to build a new **`/etc/mail/sendmail.cf`** configuration file. (If you made changes to the existing **`sendmail`** configuration file that you want to retrofit to the new configuration file, you should first save the file to another name.)

   ```
   # m4 ../m4/cf.m4 clientproto.mc > /etc/mail/sendmail.cf
   ```

HP-UX user agents are **`mail`**, **`mailx`**, and **`elm`**. **`mail`** is the standard AT&T program, whereas **`mailx`** is HP-UX's version of BSD's **`/usr/ucb/mail`**. The default routing agent is the **`sendmail`** program, and the supported delivery agents are **`/bin/rmail`** (for local users), UUCP, HP's OpenMail, X.400, and SMTP.

You have two ways to configure mail. You can:

1. Use SAM with the supplied **`sendmail`** configuration file, which provides basic connectivity appropriate for many installations.

2. Install and start it manually, as explained in the "Manually Installing sendmail" section of this chapter.

Using SAM to Install sendmail

SAM installs the default **`sendmail`** configuration file, makes **`sendmail`** executable, creates the system-wide mail alias database files from the **`aliases`** source file, and invokes the daemon.

The following steps tell how to use SAM to install sendmail:

1. Start SAM.
2. Select the **`Networking/Communications`** menu item.
3. Select **`Services Enable/Disable`**.
4. Select the **`sendmail`** menu item.
5. Select the **`Enable`** action.
6. Select **`OK`**. View the help screens if you need additional information.
7. Exit the **`Services Enable/Disable`** screen by selecting **`Exit`** from the **`List`** menu. At the **`Networking/Communications`** screen, select either **`Previous Level`** to get to a previous level, or select **`Exit SAM`** to exit from SAM.

Manually Installing sendmail

When you installed the ARPA Services product, all files and directories needed to use sendmail were installed in the correct directories with the proper permissions except the alias file (**`/usr/lib/aliases`**), the **`sendmail`** configuration file (**`/usr/lib/sendmail.cf`**), and the **`sendmail`** executable file (**`/usr/lib/sendmail`**).

To install **`sendmail`**, you must do the following:

1. Install a **`sendmail`** configuration file as **`/usr/lib/sendmail.cf`**.
2. Make **`sendmail`** executable.
3. Create system-wide mail aliases.
4. Invoke the **`sendmail`** daemon.

Installing a Configuration File

The supplied configuration file, **`/etc/newconfig/sendmail.cf`** is appropriate for many installations without modification. In particular, when initially installing sendmail, simply copying the supplied configuration file into place provides basic connectivity and permits you to integrate your system into your mail environment without immediately having to edit the configuration file. In addition, a number of localizations and routing options are provided. Descriptions of these options and detailed editing instructions are provided in the supplied configuration file itself.

To install the supplied configuration file unmodified, copy the file **`/etc/newconfig/sendmail.cf`** to **`/usr/lib/sendmail.cf`**. To make any of the supported modifications, copy **`/etc/newconfig/sendmail.cf`** to **`/usr/lib/sendmail.cf`** and edit **`/usr/lib/sendmail.cf`** according to the instructions in the file itself.

Making sendmail Executable

To make sendmail executable, as superuser, issue the command:

```
# chmod 5555 /usr/lib/sendmail
```

sendmail is normally run setuid to root (mode 5555). The default configuration is believed to be safe. However, it is possible to misconfigure **sendmail** so that it inappropriately promotes the privilege of ordinary users.

If **sendmail** does not run setuid to root, this risk is eliminated. Note that this causes sendmail to ignore the “S” mailer flag (not specified in the mailers defined in the default configuration file) and the values of the “u” and “g” configuration options, since it is unable to setuid to these users when executing mailers.

sendmail can be run non-setuid (mode 1555) if the following changes to the default configuration are made:

1. The queue directory (by default **/usr/spool/mqueue**) must be writeable by all (mode 0777).
2. The alias database (by default **/usr/lib/aliases.dir** and **/usr/lib/aliases.pag**) must be writeable by all (mode 0666). If these files do not already exist, the superuser must first create them and then **chmod** them:

   ```
   # /usr/lib/sendmail -bi

   # chmod 666 /usr/lib/aliases.dir /usr/lib/aliases.pag
   ```

Making these changes creates some security risk. Anyone will be able to delete mail from the mail queue. However, no one will be able to read other people’s mail in the mail queue.

Creating System-wide Mail Aliases

The **/etc/newconfig aliases** file is an example alias file that contains default aliases needed by sendmail. To create system-wide mail aliases, copy **/etc/newconfig/aliases** to **/usr/lib/aliases**. You can add any aliases that are appropriate for your system by editing the **/usr/lib/aliases** file. Once **sendmail** is executable issue the following command:

```
 # newaliases
```

This creates the alias database files, **/usr/lib/aliases.dir** and **/usr/lib/aliases.pag**.

Invoking the sendmail Daemon

To start the sendmail daemon, as superuser, issue the following command:

```
# /usr/lib/sendmail -bd -q30m
```

The **-bd** mode initializes the sendmail daemon to receive mail from the network. The **-q30m** flag causes **sendmail** to process the mail queue every 30 minutes.

Whenever your system is rebooted, if sendmail is executable, the **/etc/netbsdsrc** script does the following:

1. Starts the **sendmail** daemon to accept SMTP connections from the network and to process the mail queue every 30 minutes.
2. Logs the restart of the sendmail daemon in the mail log, usually **/usr/spool/mqueue/syslog**.

Stopping and Starting the sendmail Daemon

If the sendmail configuration file or frozen configuration changes, you should kill and restart the sendmail daemon so that it re-reads the configuration file.

The command

```
 # /usr/lib/sendmail -bk
```

run by the superuser, kills a **`sendmail`** daemon started with **`-bd`**, whether or not it was started with the **`-q`** interval flag. It does not kill a **`sendmail`** daemon started only with **`-q`** interval. You must kill this type of daemon by finding its process ID and killing it explicitly.

*NOTE: Do not kill **`sendmail`** with **`kill -9`**. This may cause **`sendmail`** to corrupt the alias database. Use **`kill-15`** instead.*

The **`sendmail`** frozen configuration file is **`/usr/lib/sendmail.fc`**. To freeze the configuration file, issue one of the following commands as superuser:

```
 # /usr/lib/sendmail -bz
 # /etc/freeze
```

Then restart the **`sendmail`** daemon:

```
 # /usr/lib/sendmail -bd -q30m
```

To use SAM to refreeze the configuration and kill and restart the sendmail daemon, do the following:

1. Start SAM.
2. Select the **`Networking/Communications`** menu item.
3. Select **`Services Enable/Disable`**.
4. Select the **`sendmail`** menu item.
5. Select the **`Restart`** action.
6. Fill in the form according to its instructions or answer the prompt in the window. View the help screens for information about filling in the form.
7. Exit the **`Services Enable/Disable`** screen to a previous level by selecting **`Exit`** from the **`List`** menu. Then, to exit the **`Networking/Communications`** screen, select either **`Previous Level`** to exit to a previous level or **`Exit SAM`** to exit from SAM.
8. Select apply to enter additional names of systems to be configured (use apply as a shortcut to remain in the add screen). Then, press OK when you are done with the screen.

The Mail Queue

The HP-UX mail queue directory is **`/usr/spool/mqueue`**. The naming conventions for the files in this directory are nearly identical to those found in AIX. See AIX “The Mail Queue” for more information on the file name formats.

The System Log

sendmail logs its mail messages through the **syslogd** logging facility. The **syslogd** configuration on diskless clients should forward all logging that results from sending mail to the cluster root server. The **syslogd** configuration on cluster servers and standalone systems by default write mail logging to the file **/usr/spool/mqueue/syslog**. To change this file, edit the **/etc/syslog.conf** file

Summary

AIX and HP-UX have similar mail agents in **mail** and **mailx**, respectively, but there the similarity ends. AIX's **mailx** is actually a hard link to **mail**. HP-UX's **mail** program is the old AT&T version and is not found in AIX. **elm** is packaged with HP-UX but not AIX.

Each of the two system's **sendmail** files look and feel much the same, though their locations are different. Below lists the major files and their locations.

AIX	**HP-UX**
/etc/sendmail	/usr/lib/sendmail
/etc/sendmail.cf	/usr/lib/sendmail.cf
/etc/aliases	/usr/lib/aliases
/etc/aliasesDB/DB.dir	/usr/lib/aliases.dir
/etc/aliasesDB/DB.pag	/usr/lib/aliases.pag
/etc/sendmail.cfDB	/usr/lib/sendmail.fc
/var/spool/mail	/usr/mail
/var/spool/mqueue	/usr/spool/mqueue
/usr/share/lib/Mail.rc	/usr/lib/mailx/mailx.rc
/usr/sbin/mailq	/usr/bin/mailq
/usr/sbin/mailstats	/usr/bin/mailstats
/etc/sendmail.st	/usr/lib/sendmail.st

14. X11

AIX

AIX's version of the X Window system is called AIXwindows, the 2D version of which is bundled with the operating system. The product is called AIXwindows Environment/6000 Version 1 Release 2.5, which provides support for X11R5 and binary compatibility for R4. Included with the 2D version is AIX Common Desktop Environment (CDE) 1.0.

Files and Directory Structure

AIX puts all of its X Window files in a single directory, **`/usr/lpp/X11`**, but has links to maintain compatibility with traditional X files and directories. The major links are:

```
/usr/bin/X11 -> /usr/lpp/X11/bin
/usr/lib/X11 -> /usr/lpp/X11/lib/X11
/usr/include/X11 -> /usr/lpp/X11/include/X11
```

The following standard X commands are found in **`/usr/lpp/X11/bin`**:

```
X
aixconsole
aixterm
dynamic_ext
imake
loadAbx
loadDBE
loadMbx
mkfontdir
msmit
querykbd
resize
rgb
setgamma
startx
xauth
xclock
xcmsdb
xdat
xdevicem
xdi
xhost
xinit
xinstallm
xlsfonts
xlvm
xmaintm
xmkmf
xmodmap
xpr
xprintm
xrdb
xset
xsetroot
xss
```

```
xterm
xuserm
xwd
xwud
```

The command **`mwm`** is a link to **`/usr/lpp/X11r5/Motif1.2/bin/mwm`**.

The following commands in **`/usr/lpp/X11/bin`** are actually links to files in the **`/usr/lpp/X11/Xamples`** directory:

```
appres
atobm
bitmap
bmtoa
custom
editres
listres
oclock
twm
viewres
xbiff
xcalc
xclipboard
xconsole
xcrtca
xditview
xdpr
xdpyinfo
xedit
xfd
xfontsel
xkill
xload
xlogo
xlsatoms
xlsclients
xmag
xman
xmbind
xmh
xprop
xrefresh
xsccd
xstdcmap
xwininfo
```

The **`/usr/lpp/X11/Xamples`** directory contains binaries compiled by IBM but are not officially supported. You can build new versions of these binaries as well as other utilities and demos, such as **`xev`**, by running the **`make`** utility. Complete instructions on doing so are found in **`/usr/lpp/X11/Xamples/README`**.

The location of the **`xinitrc`** and **`xserverrc`** files, as well as **`xmodmap`** key mappings are found in **`/usr/lpp/X11/defaults`**. The sample **`Xdefaults`** file is **`/usr/lpp/X11/defaults/Xdefaults.tmpl`**.

Starting X

Unless you install the **`X11.Dt`** product, he default configuration of AIX has an ASCII-type of display, called an **`lft`**, as the login terminal. You can start the X server at the shell prompt on an **`lft`** by typing **`xinit`** or **`startx`**. The older style of **`xinit`** was a script, but with X11R5 it is an executable that initializes the X server and starts **`mwm`**, an **`aixterm`**, and **`xclock`**. **`xinit`** looks for a startup file in the following order:

1. The command line
2. **`XINITRC`** variable
3. **`$HOME/.xinitrc`**
4. **`/usr/lib/X11/$LANG/xinitrc`**
5. **`/usr/lpp/X11/defaults/$LANG/xinitrc`**
6. **`/usr/lpp/X11/defaults/xinitrc`**

`startx` is a script that sets the **`DISPLAY`** variable, starts the X server, and then starts clients. **`startx`** looks for a startup file in the following order:

1. The command line
2. **`XINITRC`** variable
3. The user's **`.Xinit`**, **`.xinit`**, **`.Xinitrc`**, **`.xinitrc`**, or **`.xsession`** file

If no startup file is found, **`startx`** starts the window manager specified by the command line or indicated by the appropriate configuration file, such as **`.mwmrc`**. If no such file exists or no window manager is specified on the command line, **`startx`** starts **`mwm`** by default.

`startx` reads the resource file specified in the command line or, if not supplied, the resources from one of the following: **`.Xdefaults`**, **`.xdefaults`**, **`.Xresources`**, or **`.xresources`**.

Stopping X

You can stop the X server by selecting **`End Session`→`Quit…`** from the **`Root Menu`** or by pressing the **`Ctrl`**+**`Alt`**+**`Backspace`** key combination.

aixterms

The default terminal emulator on AIX is an **`aixterm`**, which emulates a IBM's high function terminal. If you **`telnet`**, **`rlogin`**, or **`rsh`** to a machine that does not have **`aixterm`** terminal definitions—and HP-UX machines are among those—you can set the TERM variable to **`vt100`**, **`vt102`**, **`vt220`**, **`vt320`**, or **`xterm`** to use programs like **`vi`** properly, or you can add **`aixterm terminfo`** entries by doing the following:

1. On the AIX box, **`cd`** to **`/usr/share/lib/terminfo/a`**.
2. Issue the following command:

   ```
   # infocmp aixterm > /tmp/aixterm.ti
   ```

3. On the HP-UX box, **`cd`** to **`/tmp`**.

4. **`ftp`** to the RS/6000.
5. While in **`ftp`** change directories to **`/tmp`**.
6. Do a **`get`** on **`aixterm.ti`**.
7. Quit **`ftp`**.
8. Back on the HP machine, type the following command:

   ```
   # tic aixterm.ti
   ```

This procedure will put an **`aixterm`** terminfo entry into the HP's **`/usr/lib/terminfo/a`** directory.

For purposes of interoperability some people like using **`xterm`**s in order to avoid procedures like the above. However, there are problems in doing so. For example, if you run the **`smitty`** command in an **`xterm`** instead of an **`aixterm`** you will not have all the normal function keys available to you. This is also true for the **`smit`** command run remotely via **`telnet`**, **`rlogin`**, or **`rsh`**. Instead, you have to use escape sequences for certain functions. Also, the **`tn3270`** command works best in an **`aixterm`**. If you use the Common Desktop Environment, then a **`dtterm`** is your best interoperability solution.

Making the RS/6000 a Font Server

Before you can create a font server, you must install the **`X11.fnt.fontServer`** product (separate from BOS installation). AIX's generic font server configuration file is **`/usr/lpp/X11/lib/X11/fs/config`**. If you **`cd`** to the linked directory **`/usr/lib/X11/fs`** you will find it as well. This file is simple and similar to those found on other platforms except that it stipulates port 7500 instead of 7000. Any free port will do, but just remember the correct number when you run the **`xset`** command on the font client.

To start the font server type:

```
 # fs -config /usr/lpp/X11/lib/X11/fs/config &
```

To get the font client to include the font server fonts in its font path:

```
 # xset +fp tcp/server_name:7500
```

xdm

If you want to use X Display Manager, you must load the **`X11.apps.xdm`** product and configure the system with the **`xdmconf`** utility. This utility, along with all the other **`xdm`**-related files are found in **`/usr/lpp/X11/lib/X11/xdm`**. For compatibility's sake there is a link from **`/usr/lib/X11/xdm`** to this directory.

`xdmconf` will configure **`xdm`** to run at bootup via the SRC if TCP/IP is set up. The SRC command to start **`xdm`** is placed at the end of **`/etc/rc.tcpip`**. If TCP/IP is *not* set up for your machine, then **`xdm`** will be placed in **`/etc/inittab`** for startup by **`init`**. To remove **`xdm`** configuration, type the following:

```
 # cd /usr/lib/X11/xdm ; ./xdmconf -d
```

HP-UX

The HP-UX 10 X11 server is based on the X Consortium's X11 Revision 6 sample implementation, augmented with HP modifications. Although it is based is R6-based, the X server has been built with the R5 version of the X11 library. A full R6 X11 implementation will be available in a later release.

Hewlett-Packard does not provide or support the entire core MIT distribution. Many of these programs or clients are sample implementations or perform tasks that are accomplished by other clients in the Common Desktop Environment. A number of unsupported core MIT clients and miscellaneous utilities are provided in **`/usr/contrib/bin`**. In addition, the entire core MIT distribution, compiled for Hewlett-Packard platforms, can be obtained from HP's users group INTERWORKS for a nominal fee.

Files and Directory Structure

The following standard directories for X binaries and configuration files are provided:

```
/usr/bin/X11
/usr/lib/X11
```

Shared libraries are delivered in subdirectories with names specific to their release:

```
/usr/lib/Motif1.1
/usr/lib/Motif1.2
/usr/lib/Motif1.2_R6
/usr/lib/X11R4
/usr/lib/X11R5
/usr/lib/X11R6
```

As you can see, X11R6 libraries are shipped with X11R5 libraries. The default build environment for Motif is version 6.

HP-UX 10.20 includes both the X11R5 and R6 versions of **`xterm`**, delivered in the following directories:

```
R5: /usr/bin/X11/xterm
R6: /usr/contrib/bin/X11/xterm
```

The functionality of **`xterm`** is superseded by the industry-standard **`dtterm`**.

The executable files and man pages for all the unsupported clients are installed in **`/usr/contrib/bin/X11`** and **`/usr/contrib/man`**, respectively. In order to use these clients, be sure that your PATH environment variable contains **`/usr/contrib/bin/X11`**. Use **`env`** to see the current setting of your environment variables. The online **`man`** page access utility looks in **`/usr/contrib/man`** by default, so you don't need to modify the MANPATH environment variable.

Unsupported Clients

```
appres
bitmap
dr_dt
editres
fontname
fontprop
```

```
fsinfo
fslsfonts
iceauth
keymap_ed
oclock
showrgb
smproxy
startx
twm
x11start
xclipboard
xconsole
scutsel
xdpyinfo
xfd
xinit
xkbcomp
xkill
xload
xlogo
xlsatoms
xlsclients
xmag
xmh
xon
xpr
xprop
xrefresh
xstdcmap
xterm
xwininfo
```

Starting X

On HP systems, the X Window System is normally started via the Common Desktop Environment (CDE), which uses **`xdm`** (X Display Manager) technology rather than **`xinit`** to start X. When CDE is not used, the normal method of starting X is via the **`x11start`** script which is simply a "wrapper" around **`xinit`** providing environment and command line setup appropriate for HP systems. Currently this process is not supported but is provided for compatibility to previous setups.

Specifically, **`x11start`** performs the following startup tasks:

- PATH environment variable set-up appropriate for the X environment
- X server startup
- Selected client(s) startup from a specific client file
- General user resource loading from a specific resource file

`x11start` encompasses the following components:

- The **`/usr/contrib/bin/X11/x11start`** script
- The **`/usr/contrib/bin/X11/xinit`** program
- The default client script, **`/usr/contrib/lib/X11/sys.x11start`**

- The default resource file, **`/usr/contrib/lib/X11/sys.Xdefaults`**

Customized client and resource files can be created by copying the default files to **`$HOME/.x11start`** and **`$HOME/.Xdefaults`**, respectively, and then customizing them. Customized files that exist (with the appropriate permissions) will be used by the **`x11start`** components in place of the default files.

Client options pass from the **`x11start`** command line to all clients in the **`.x11start`** file that have a **`$@`** parameter. The options replace the parameter. This method is most often used to specify a display other than the usual one on which to display the client. You can, however, use the command-line option to specify a non-default parameter, such as a different background color.

The default **`.x11start`** file starts the following clients:

- A terminal emulation client, such as **`hpterm`**
- **`mwm`**

Server options are preceded with a double hyphen (**`--`**). If the option following the double hyphen begins with a slash (**`/`**) or a path and a slash, it starts a server other than the default server. If the option begins with a colon followed by a digit (**`:#`**), it specifies the display number (**`0`** is the default display number). Additional options specified after the server or display refer to the specified server or display. See the XSERVER page in the reference section for more information on server options.

Examples

```
# x11start
```

This starts the default **`xinit`** server and executes the client script (either **`/usr/contrib/lib/X11/sys.x11start`** or **`$HOME/.x11start`**) without passing arguments to either.

```
# x11start -bg black -fn 24x36
```

This starts the default **`xinit`** server and executes the client script, passing it all of the arguments. If the default client script is used, **`-bg black -fn 24x36`** is passed to both **`mwm`** and **`hpterm`** since both of their default command lines contain **`$@`**. If the default client script is used, the actual **`xinit`** command executed is **`xinit /usr/contrib/lib/X11/sys.x11start -bg black -fn 24x36.`**

```
# x11start -fg white -- :1
```

This starts the default server on display 1 and executes the client script with the arguments, **`-fg white`**. If the customized client script is used, the **`xinit`** command line is **`xinit $HOME/.x11start -fg white -- :1`**.

```
# x11start -- Xhp -a2 -t 5
```

This starts the server, **`Xhp`** with the arguments, **`-a2 -t 5`** and then executes the client script without any arguments.

Stopping X

After stopping all application programs, stop the window system by holding down the **`Ctrl`** and **`Left Shift`** keys, and then pressing the **`Reset`** (**`Pause-Break`**) key. This stops the display server, and with it the window system.

Fonts

HP-UX's font server, **`xfs`**, is R6-based. The file **`/usr/bin/X11/fs`** is actually a link to **`xfs`**. You can configure a font server (or client) by running the following interactive script:

```
# /sbin/set_parms font_c-s
```

Or you run the following noninteractive command:

```
# /usr/sbin/mk_fnt_srvr
```

These commands will start the font server as defined in **`/sbin/init.d/xfs`**, which is:

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

They will also set the RUN_X_FONT_SERVER variable in **`/etc/rc.config.d/xfs`** to 1 so that font services take effect at system startup.

To remove font server configuration:

1. Edit **`/etc/rc.config.d/xfs`** and set the RUN_X_FONT_SERVER variable to 0.
2. Issue the following command:

   ```
   # /sbin/init.d/xfs stop
   ```

According to the **`xfs man`** page, there is no default configuration file. Instead, you must specify such a file with the **`-config`** ***`file`*** option. Likewise, there is no default port: it must be specified either on the command line or in the configuration files.

If the font server is to be run on the same machine as the X server and if the font server is started before the X server, the font server does not need to be added to the font path of the X server. However, if the font server is started after the X server is already running, or if a font server is running on a different machine than the X server, then the font server must be added to the font path of the X server as follows:

```
# xset +fp tcp/:7000
```

This adds the font server port to the front of the font path list searched by the X server. If you are adding a font server that is not running on the same machine as the X server, the **`xset`** command is extended to also specify the font server host:

```
# xset +fp tcp/host_name:7000
```

Once the font server is running and has been added to the font path of an X server, the server must rehash its fonts. This is also done with the **`xset`** client via:

```
# xset fp rehash
```

hpterms

The **`hpterm`** client emulates a Term0 terminal.   The syntax of the hpterm client is as follows:

```
# hpterm [-options] [&]
```

There are too many options to cover here. Refer to the **`hpterm man`** page for all the options available.

When running block mode applications, it may be necessary for **`hpterm`** to identify itself to application programs as some terminal other than "X-hpterm." Most applications understand the terminal id "2392A." Newer applications also understand the terminal id "700/92" while older applications may only understand the terminal id "2622A." To set the terminal identification string, use the **`-ti`** command line option, the **`termId`** resource, or the **`TermId`** class.

To put **`hpterm terminfo`** entries on an RS/6000:

1. **`cd`** to **`/tmp`**
2. Type the following command to put the **`hpterm terminfo`** entry into an ASCII file:

   ```
   # untic hpterm > hpterm.ti
   ```
3. Put the **`hpterm.ti`** file on the RS/6000:

   ```
   ftp RS_node

   ftp> cd /tmp

   ftp> put hpterm.ti

   ftp> quit
   ```
4. On the RS/6000, compile the **`hpterm.ti`** file:

   ```
   # tic /tmp/hpterm.ti
   ```

This will put the appropriate **`terminfo`** entries in **`/usr/share/lib/terminfo/h`** on the RS/6000.

X11 Interoperability

X, like TCP/IP, NFS, NIS, DNS, and sendmail, is essentially the same on both AIX and HP-UX. Only the details differ. X has made interoperability among UNIX systems much easier in that clients can run on one platform and display on another. This is even true of clients such as SAM and SMIT, which you may find necessary to access from a box of another ilk.

Displaying to a Remote Host

Using **`xhost`**, you can add or delete a remote host's permission to access the local display server. For example, the following command allows the remote host **`stargate`** to access your local display.

```
 # xhost  +stargate
```

The default display on which a client is displayed is obtained from the DISPLAY environment variable of the system on which the client starts. It sets the host, display number, and screen number to which the client directs its output. This is typically display 0, screen 0. Most clients have a **`-display`** option that lets you set the host, display number, and screen on which the client will display its output. The **`-display`** option has the syntax:

```
 -display [host:display.screen]
```

where:

`host` is the hostname of a valid system on the network.

`display` is the number of the display on the system on which you want the output to appear. A display can include more than one screen.

`screen` is the number of the screen where the output is to appear. The default is 0.

For example, executing the command:

```
 # hpterm -display rs_node:0.1 &
```

starts an **`hpterm`** process on the local system and displays the window on display 0, screen 1 of the ***`rs_node`*** system. The window has the default size, location, and color. Conversely, you can run the following command on an RS/6000:

```
 # aixterm -display hp_node:0 &
```

If you need the capabilities of these terminal emulators on machines other than the ones they were designed for, this is the way to do it. However, the HP machine will not have the proprietary IBM fonts that will give the **`aixterm`** the same appearance it has when run locally on an RS/6000. The same is true for RS/6000s displaying **`hpterms`**. This fact does not hinder the functionality of these emulators, but if you prefer for each to look exactly the same whether running locally or remotely on another platform, you must do one of two things:

- Make one machine from each platform a font server and put the font server in the font path of the client with the **`xset`** command. Details for font servers are described above.

There are several ways to run programs on a remote host from a command line:

- Use **`rlogin`** to log into the remote host.
- Use **`remsh`** to start a client remotely without formally logging in.

If the client produces output on a display, you must specify the display and screen on which you want the output to appear.

Running Programs Using rlogin

You can use an existing terminal emulator window to log into a remote host. Once the window is acting as a terminal off the remote host, you can run clients there and direct the output to any display. For example, the following commands log into and start **`xload`** on remote host ***`rs_node`*** and display the output on local system ***`hp_node`***.

```
 # rlogin rs_node
 # xload -display hp_node:0.0 &
```

Using Remote Shells to Start Programs

The benefit of using **`remsh`** or **`rsh`** instead of **`rlogin`** is that the local system starts only one process (the client) with a remote shell; with the remote login, the local system starts both the remote login and the client.

The following syntax starts a remote shell on a remote host, redirects **`remsh`** input, starts a client, and directs output to the local display.

```
 # remsh remote -n client -display local:display.screen &
```

where:

`remote` is the remote host name.

`client` is the absolute path of the executable client file (**`remsh`** does not allow the PATH variable).

`local` is the local host name.

For example, the following command runs **`xload`** on remote host ***rs_node*** and directs output to the display of system ***hp_node***.

```
 # remsh rs_node -n /usr/bin/X11/xload -display hp_node:0.0 &
```

Going the other way:

```
 # remsh hp_node -n /usr/bin/X11/xload -display rs_node:0.0 &
```

NOTE: To make these examples run flawlessly requires use of the **`xhost`** command and the appropriate **`.rhosts`** entries. For more information on executing remote commands without a password check the **`remsh man`** page.

Turning Your Workstation into an X Terminal

Occasionally you might want to connect to a machine of a different platform and perform administrative tasks completely within the environment of the remote machine. If you do, all you need to do is convert your local workstation into an X terminal connected to the remote machine.

To convert an HP workstation into an X terminal running off an RS/6000 called ***rs_node***:

1. Make sure CDE is running on the RS/6000.
2. Stop the X Window system on the HP by choosing a **`No Windows`** login from the login screen.
3. Log in to the ASCII display.
4. Type the following command:

   ```
   # X -query rs_node
   ```

5. At this point you will see an RS/6000 CDE session running on your HP. Login as you normally do.
6. To stop your session choose **`End Session`** from the Root Menu.
7. To stop your X server on the HP press the **`Ctrl`**+**`Left Shift`**+**`Pause`** key combination. You will return to your HP ASCII screen. Type **`exit`** to resume your HP CDE session.

To convert an RS/6000 workstation into an X terminal running off an HP 9000 called ***hp_node***:

1. Make sure CDE is running on your HP.

2. Login to your **`lft`**. If X starts up automatically, stop it by pressing **`Ctrl`**+**`Alt`**+**`Backspace`**.

3. Type the following command:

   ```
   # X -query hp_node
   ```

4. At this point you will see HP CDE appear on your RS/6000. Login as you normally do.

5. Logout as you normally do on an HP system.

6. To stop the X server, press **`Ctrl`**+**`Alt`**+**`Backspace`**.

CDE Interoperability

Among the advantages of using CDE on both AIX and HP-UX systems is that it provides a mechanism for effective client-server operations. By using the **`desktop subprocess control`** daemon (**`dtspcd`**) you can have one machine act as an application server and another act as the session server and display. With the X Window system you can use the **`xhost`** and **`remsh`** mechanism, but using **`dtspcd`** streamlines this process.

Essentially you create an **`action`** that executes a program remotely but displays locally. An action is a way to start an executable (a script, command, application, etc.) by clicking on an icon. Actions are defined in ASCII database files with **`.dt`** extensions to their file names and can be found in:

```
$home/.dt/types
/etc/dt/appconfig/types/$LANG
/usr/dt/appconfig/types/$LANG
```

The easiest way to create an action is to use the **`Create Action`** icon. Once you create your action you must edit its **`.dt`** file so that the EXEC_HOST field exists. For example, if you create an action called **`info`** on an HP-UX system so that it runs InfoExplorer on an RS/6000 when you click on the **`info`** icon, that action definition may look like:

```
ACTION info
{
      LABEL             info
      TYPE              COMMAND
      EXEC_HOST         hpbarr2
      EXEC_STRING       /usr/bin/info
      ICON              Dtactn
      WINDOW_TYPE       NO_STDIO
}
```

This specifies that clicking on the **`info`** icon will start **`/usr/bin/info`** on **`hpbarr2`**, which happens to be an RS/6000 running AIX. All of this is possible if the following conditions are met:

1. That user names and UIDs are consistent across systems (commonly done with NIS).

2. That home directories are shared by all client and server systems (usually done with NFS). This is required for **`dtspcd`** authentication and X authorization. NFS mounts can be static or automounted.

3. That **`dtspcd`** is properly configured.

To configure **`dtspcd`**:

1. Enable the service in both **`/etc/inetd.conf`** and **`/etc/services`**.
2. If you are using automounter, you must use the **`-mount_point`** option in **`/etc/services`** if you are not using the default **`/net`** mount point. For example:

   ```
   dtspcd stream tcp nowait root /usr/dt/bin/dtspcd \
   /usr/dt/bin/dtspcd -mount_point /u
   ```

3. If you don't use home directories for authentication, you must specify the alternate mounted directory with the **`-auth_dir`** option in **`/etc/services`**.
4. Use the **`-debug`** or **`-log`** options to help troubleshoot the problem.
5. For HP-UX systems enable **`dtspcd`** in **`/var/adm/inetd.sec`**.

For more information on **`dtspcd`** see *Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide*.

Appendix A

File and Directory Comparisons

Directory Mappings

HP-UX 9.X	AIX 4.1.X	HP-UX 10.X
/dev	/dev	/dev
/etc	/etc	/etc
-	/export	/export
-	/mnt	/mnt
/users	/home	/home
-	/sbin	/sbin
/tmp	/tmp	/tmp
/usr	/usr	/usr
-	/usr/ccs	/usr/ccs
/usr/include	/usr/include	/usr/include
-	/usr/lbin	/usr/lbin
/usr/lib	/usr/lib	/usr/lib
-	/usr/sbin	/usr/sbin
-	/usr/share	/usr/share
/usr/man	/usr/share/man	/usr/share/man
-	/var	/var
/usr/adm	/var/adm	/var/adm
/usr/spool	/var/spool	/var/spool
/usr/tmp	/var/tmp	/var/tmp
/usr/mail	/var/spool/mail	/var/mail
/usr/local	-	/usr/local
/etc/newconfig	-	/usr/newconfig

System Files

HP-UX 9.X	AIX 4.1.X	HP-UX 10.X
/etc/netlinkrc	/etc/rc.tcpip	/etc/rc.config.d/netconf
/etc/exports	/etc/exports	/etc/exports
/etc/disktab	-	/etc/disktab
/etc/checklist	/etc/filesystems	/etc/fstab
/etc/passwd	/etc/passwd, /etc/security/passwd	/etc/passwd
/etc/group	/etc/group, /etc/security/group	/etc/group
/etc/hosts	/etc/hosts	/etc/hosts
/etc/hosts.equiv	/etc/hosts.equiv	/etc/hosts.equiv
/etc/mnttab	/etc/filesystems	/etc/mnttab
/usr/spool/lp/interface/*	/etc/qconfig	/etc/lp/interface/*
/etc/rc	/etc/rc	/sbin/rc
/etc/resolv.conf	/etc/resolv.conf	/etc/resolv.conf
/etc/services	/etc/services	/etc/services
-	/etc/netsvc.conf	/etc/nsswitch.conf
/etc/xtab	/etc/xtab	/etc/xtab
/hp-ux	/usr/lib/boot/unix_up	/stand/vmunix

General Commands

HP-UX 9.X	AIX 4.1.X	HP-UX 10.X
/usr/bin/bdf	/usr/bin/df -k	/usr/bin/bdf
/bin/ls	/usr/bin/ls	/usr/bin/ls
/etc/ioscan	/usr/sbin/lsdev -C	/sbin/ioscan
/etc/diskinfo	-	/usr/sbin/diskinfo
/bin/uname -I	/usr/sbin/hostid	-
/usr/bin/sam	/usr/bin/smit	/usr/sbin/sam
/usr/bin/sar[10]	/usr/sbin/sar	/usr/sbin/sar
/usr/bin/lp	/usr/bin/enq /usr/bin/lp /usr/bin/lpr /usr/bin/qprt	/usr/bin/lp
/usr/bin/lpstat	/usr/bin/enq -A /usr/bin/lpq /usr/bin/lpstat /usr/bin/qchk	/usr/bin/lpstat
/usr/bin/cancel	/usr/bin/cancel /usr/bin/lprm /usr/bin/qcan /usr/bin/enq -x	/usr/bin/cancel
/usr/bin/remsh	/usr/bin/remsh /usr/bin/rsh	/usr/bin/remsh
/usr/etc/automount	/usr/sbin/automount	/usr/sbin/automount
/etc/gated	/usr/sbin/gated	/usr/sbin/gated
/usr/bin/netstat	/usr/sbin/netstat	/usr/bin/netstat
/etc/route	/usr/sbin/route	/usr/sbin/route
/etc/ping	/usr/sbin/ping	/usr/sbin/ping
/etc/ypbind	/usr/lib/netsvc/yp/ypbind	/usr/lib/netsvc/yp/ypbind

[10] 800 series only.

/usr/bin/ypcat	/usr/bin/ypcat	/usr/bin/ypcat
/usr/bin/ypwhich	/usr/bin/ypwhich	/usr/bin/ypwhich
/usr/bin/ypmatch	/usr/bin/ypmatch	/usr/bin/ypmatch
/usr/bin/yppasswd	/usr/bin/yppasswd	/usr/bin/yppasswd

LVM Commands

HP-UX 9.X	AIX 4.1.X	HP-UX 10.X
-	/usr/sbin/chpv	/usr/sbin/pvchange
-	/usr/sbin/mkdev -c	/usr/sbin/pvcreate
-	/usr/sbin/lspv	/usr/sbin/pvdisplay
-	/usr/sbin/migratepv	/usr/sbin/pvmove
-	/usr/sbin/mkvg	/usr/sbin/vgcreate
-	-	/usr/sbin/vgremove
-	/usr/sbin/chvg /usr/sbin/varyonvg /usr/sbin/varyoffvg	/usr/sbin/vgchange
-	-	/usr/sbin/vgcfgrestore
-	/usr/sbin/exportvg	/usr/sbin/vgexport
-	/usr/sbin/importvg	/usr/sbin/vgimport
-	/usr/sbin/lsvg	/usr/sbin/vgscan
-	/usr/sbin/extendvg	/usr/sbin/vgextend
-	/usr/sbin/reducevg	/usr/sbin/vgreduce
-	/usr/sbin/chlv	/usr/sbin/lvchange
-	/usr/sbin/lspv	/usr/sbin/lvdisplay
-	/usr/sbin/extendlv	/usr/sbin/lvextend
-	-	/usr/sbin/lvreduce
-	/usr/sbin/rmlv	/usr/sbin/lvremove
-	-	/usr/sbin/lvlnboot
-	-	/usr/sbin/lvrmboot
-	/usr/sbin/chfs	/usr/sbin/extendfs
-	-	/usr/sbin/lvsplit
-	-	/usr/sbin/lvmerge