HEWLETT® PACKARD

HP 9000
Series 300/400
Computers

Managing Clusters of HP 9000 Computers: Sharing the HP-UX File System

Managing Clusters of HP 9000 Computers Sharing the HP-UX File System

HP 9000 Series 300 and 400 Computers

HEWLETT PACKARD

HP Part No. B1864-90015
Printed in USA August 1992

First Edition
E0892

Legal Notices

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

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

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

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

Printing History

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

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

First Edition: August 1992 (HP-UX Release 9.0)

Contents

Figures

Tables

1

Introduction

If you administer, or are about to administer, an HP-UX cluster with a Series 300 or 400 root server, this book is for you. (If your cluster has a Series 700 server, you need a different version of this manual, part number B2355-90038.) If you're not sure what a cluster is, this chapter will tell you—see "What is a Cluster?" a little further on.

A cluster is similar in many ways to a single, multi-user system, and if you have administered such a system, you'll find that much of your experience is applicable to managing a cluster.

By the same token, this book is not intended to replace the *System Administration Tasks* manual. Administration tasks that are the same in a cluster as they are on a single workstation or multi-user computer are not covered here. This book concentrates on those tasks, or parts of tasks, that are unique to clusters.

How To Use this Book

The following table lists some of the tasks involved in creating and managing a cluster and shows where to start. References in italics are to other HP manuals supplied with the system. References to chapters are to this manual.

To accomplish this	Do this
Understand what a cluster is and how it works	Read the rest of this chapter and Chapter 2, "Understanding Clusters"
Plan a Local Area Network to support the cluster	Use Chapter 3, "Setting Up the Network" in conjunction with *Installing and Administering LAN/9000*.
Decide how to distribute disk drives and other peripherals in the cluster	Read Chapter 12, "Adding Peripherals to a Cluster".
Create a cluster starting with computers in boxes	Follow directions in Chapter 4, "Setting Up a Cluster".
Convert a running system to a cluster server	Follow directions in Chapter 4, "Setting Up a Cluster".
Convert a running system to a cluster client	Follow directions for adding a client in Chapter 5, "Adding Cluster Clients".
Boot and shut down the cluster	Follow procedures in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
Manage cluster users	Read Chapter 13, "Managing Users in a Cluster".
Back up the cluster	Use special options to HP-UX commands shown in Chapter 9, "Backing Up Files in a Cluster".
Install or move disk drives, tape drives, etc.	Use Chapter 12, "Adding Peripherals to a Cluster" in conjunction with *Installing Peripherals*.
Connect the cluster to another network	Follow directions in Chapter 6, "Connecting the Cluster to Another Network".

To accomplish this	Do this
Reconfigure the kernel on a cluster node (server or client)	Use Chapter 11, "Reconfiguring the Kernel for a Cluster Node" in conjunction with the *System Administration Tasks* manual.
Evaluate or design application software for use in the cluster	Read Chapter 15, "Applications in a Cluster".
Install or update application software	Follow directions in Chapter 14, "Updating a Cluster".
Update the cluster to a new release of HP-UX	Follow directions in Chapter 14, "Updating a Cluster".
Check for any changes in routine tasks caused by being in a cluster	Read Chapter 8, "Introduction to Cluster Administration".
Remove a cluster client	Follow directions in Chapter 7, "Removing and Renaming Cluster Clients".

What is a Cluster?

An HP-UX cluster is a means of sharing resources, and particularly the file system, among HP 9000 computers in a network. This manual is concerned only with clusters of Series 300 and 400 computers.

A cluster consists a network of HP 9000 computers, connected by **LAN (Local Area Network)** hardware and software, in which only one computer has a root file system. This computer is known as the **cluster root server** (sometimes shortened to **cluster server** and occasionally to **server**).

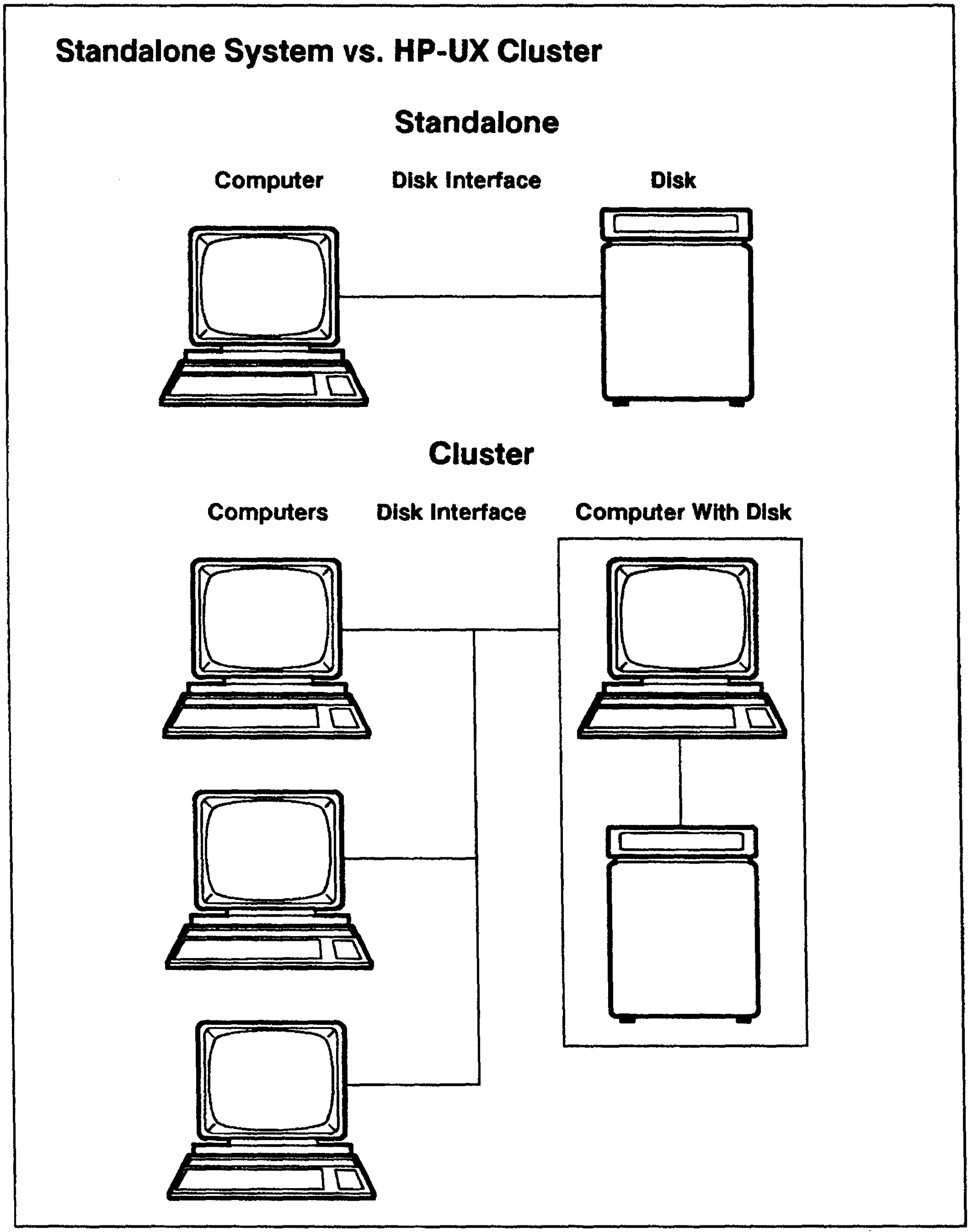
The cluster root server shares the root file system with all the other machines in the cluster, which need not have any disks of their own and are called **cluster clients**, or occasionally **clients**.

The cluster server can also function as a workstation or multi-user system: it is not restricted to servicing file-system requests.

Each machine in a cluster (the cluster server or any client) can be referred to as a **cluster node**, occasionally shortened to **cnode**.

It is possible for cluster clients to have their own disk drives. These can hold file systems (but not the root file system, which must be on the cluster server) or swap space, or both.

The illustration on the next page shows how a cluster differs from a single workstation or multi-user system.



LG200151_001

Cluster vs. Workstation and Multi-User Systems.

HP-UX clusters are designed to combine the advantages of desktop workstations on one hand, and single, multi-user systems on the other.

As with a multi-user system, file and swap space is shared, and so is the line-printer spooler (which allows a printer attached to any node in the cluster to accept print requests from any client as well as from the server, and also supports remote spooling to a computer outside the cluster.) The behavior of each of the major kinds of peripheral in a cluster is documented in Chapter 12, "Adding Peripherals to a Cluster".

On the other hand, a cluster user retains the major advantage enjoyed by a workstation user: access to each computer's processor is either exclusive or shared with a small group of terminal users.

You'll find a more extended comparison between workstations, multi-user systems and clusters near the beginning of Chapter 2, "Understanding Clusters".

Building And Running A Cluster

What's Involved in Setting Up a Cluster?

Creating a cluster breaks down into the following steps:

1. Decide what kind of cluster you want.

   See "Different Kinds of Cluster", in Chapter 2, "Understanding Clusters".

2. Make sure you have all the hardware and software you need.

   See "What You Need Before You Start", in Chapter 4, "Setting Up a Cluster".

3. Set up the network.

   See Chapter 3, "Setting Up the Network"

4. Gather information needed to configure the cluster server.

   See "Gathering Information" in Chapter 4, "Setting Up a Cluster".

5. Configure the cluster server.

   You'll use the **SAM** (System Administration Manager) utility to do this.

   See "Configuring the Cluster Server" in Chapter 4, "Setting Up a Cluster".

6. Add Cluster Clients.

   You'll use the **SAM** (System Administration Manager) utility to do this.

   See Chapter 5, "Adding Cluster Clients".

What's Involved in Administering a Cluster?

Administering a cluster is similar in principle to administering a single, multi-user system: in both cases you must balance a pool of resources against the needs of a group of users.

In a cluster, however, you are managing several distinct computers, even though they share the file system and other resources. This changes many routine tasks: you need special HP-UX commands, or special options of ordinary commands; or it makes a difference which node you are on when you perform the task (when you modify a client's kernel, for example, you must be logged in to that client).

The following chapters explain how to do tasks that are changed because you are in a cluster:

- Chapter 8, "Introduction to Cluster Administration"
  - Explains the ways routine tasks are changed in a cluster and lists cluster-specific commands and options.
- Chapter 9, "Backing Up Files in a Cluster"
  - Provides sample cluster-wide backup commands.
- Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes"
  - Explains cluster boot and shutdown
- Chapter 11, "Reconfiguring the Kernel for a Cluster Node"
  - Provides guidelines for reconfiguring a cluster node's kernel.
- Chapter 12, "Adding Peripherals to a Cluster"
  - Explains the rules governing classes of peripherals in a cluster, provides information to help you decide how to distribute peripherals, and gives examples for adding each each class of peripheral to the server and/or to a client.
- Chapter 13, "Managing Users in a Cluster"
  - Discusses the issues affecting users of a cluster, as opposed to users of workstations or single, multi-user systems.

- Chapter 14, "Updating a Cluster"
  - Explains how to install and update software in a cluster.
- Chapter 15, "Applications in a Cluster"
  - Discusses issues affecting application software in a cluster.

2

Understanding Clusters

Most end-users find clusters very easy to work with and notice little if any difference from a workstation or multi-user system. However, this simple interface hides some architectural complexity which you, as the administrator of the cluster, must be aware of. This chapter describes in detail what an HP-UX cluster is and how it works.

Why Use a Cluster?

There are two conventional types of computer systems: timeshared systems consisting of a single mainframe or mini-computer with terminals, and systems consisting of personal workstations connected by local area networks. Each type of system has its own advantages and disadvantages. These are outlined in Table 2-1 and Table 2-2.

Table 2-1. Advantages/Disadvantages of Timeshared Systems

Advantages	Disadvantages
transparent peripheral sharing	performance limitations: as number of users goes up, performance goes down.
transparent file sharing	limited reconfiguration ability
same login from any terminal	limited human interfaces (few or no bit-mapped displays)
no duplication of shared resources	large initial investment
only one system administrator required	

Table 2-2. Advantages/Disadvantages of Networked Workstations

Advantages	Disadvantages
better performance for each user	files not guaranteed to have same names on different systems
support bitmapped graphics and windows displays	remote files must be accessed with different commands and system calls than local files
provide for incremental growth	performance is often significantly lower when accessing a remote resource
lower initial investment	each user must perform system administration functions for his or her own workstation
	significant number of duplicate files on each workstation

Advantages of a Cluster

An HP-UX cluster combines the advantages of a timeshared system with the advantages of networked workstations. Some of the advantages are:

- Same view of a global file system from each workstation.
- Single point system administration—only one system administrator required for a cluster.
- Flexibility of configuration.
- Dynamic reconfiguration.
- High performance for individual workstations.
- Sharing of costly resources.
- A bit-mapped display per user becomes practical.

What Is a Cluster?

A **cluster** consists of two or more workstations linked together by a local area network (LAN) but having only one root file system. From the point of view of the file system, all the machines appear as one system. From the point of view of processors and processing space, each machine in the cluster is distinct.

A cluster consists of a **cluster root server** and one or more **cluster clients**. Each computer in the cluster (including the cluster root server) is referred to as a **cluster node**, sometimes shortened to **cnode**.

The **cluster root server** (sometimes referred to as the **cluster server** or **root server**, or occasionally **server**) is the cluster node that has the root file system.

The cluster root server supports other workstations (the cluster clients), which may, but need not, have their own disks. All cluster clients boot, over LAN, from kernels residing in the cluster server's file system. Each client has its own kernel.

A cluster client that has no local disks is known as a **diskless client** or **diskless node**. A cluster client that has one or more local disks that other clients are sharing is known as an **auxiliary swap server** (or **swap server**), or an **auxiliary file server**, or generically as an **auxiliary server**—depending on how the disks are being used.

All cluster nodes (the server and all the clients) have access to files on the cluster root server's and any auxiliary file server's disks.

Cluster clients usually swap to the cluster server's swap area. However, a cluster client can instead swap to its own local disk, or to an auxiliary swap server's disk.

Chapter 12, "Adding Peripherals to a Cluster" provides rules and guidelines for distributing disks between the root server and the clients.

Setting Up Cluster Hardware

Peripherals

File system disks can be mounted on any **cluster node** (that is, on any computer in the cluster—server or client) but the root file system and any other file systems containing HP-UX system software must be mounted on the root server.

Your backup device should also be connected to your cluster server. Modems for UUCP *must* be on the cluster server.

The cluster server or any cluster client can have spooled printers and plotters which are shared by all members of the cluster.

A cluster client can also have local devices such as HP-IB instruments, non-spooled printers and plotters, and tape drives, but these local devices can be used only by processes on that cluster client. (Cluster nodes can invoke processes on other cluster nodes with the `remsh` and `rlogin` commands).

Chapter 12, "Adding Peripherals to a Cluster" contains detailed rules, guidelines and instructions for connecting peripherals to a cluster.

LAN

All nodes in the cluster must be connected by a single LAN. You can have more than one cluster per LAN, and computers that are not part of any cluster can also be on the cluster's LAN (we'll use the term **standalone** from now on to describe computers that are not part of any cluster).

Hewlett-Packard recommends that the cluster be on a small local LAN, with the root server acting as a gateway to other networks. This way, there will be less contention, and therefore better performance.

Chapter 3, "Setting Up the Network" and Chapter 6, "Connecting the Cluster to Another Network" contain more detailed recommendations and guidelines.

Sample Cluster

In Figure 2-1 the cluster root server (the cluster node with the root file system) is called `server`. There are two cluster clients: `client1` and `client2`. `server` has a second LAN card and is a networking gateway to computers not on the cluster's LAN.

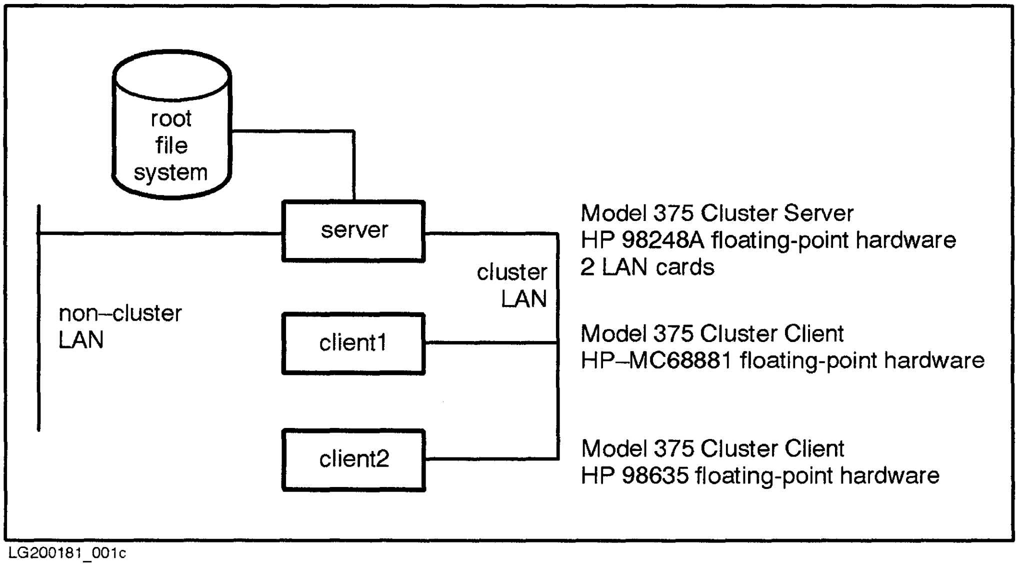


Figure 2-1. Sample Cluster

Administering a Cluster

You can perform most system administration tasks from the cluster server, and much of what you do will be the same as what you would do on a standalone system, because the entire cluster uses a single, "global" file system.

There are some important differences, however. In some cases the difference lies in where (on which cluster node) you perform a task. In other cases the task itself is different: either something you do only in a cluster, or something that is significantly changed because you are doing it in a cluster.

Tasks Restricted to Specific Cluster Nodes

There are some tasks you must perform while logged in to a specific cluster node.

For example, you must be logged in to the cluster server to add cluster clients, but if you need to modify a client's kernel, you must be logged in to that particular client.

You can still accomplish such tasks from the root server or another client by using the **rlogin** and **remsh** commands. For example, if you are on the cluster server, and want to create a kernel for the cluster node called `client2`, you could either walk over to `client2` and log in there, or execute the following series of commands from the server:

`rlogin client2`

You will now see the copyright messages on "client2".

When you see a system prompt, type:

`/usr/bin/sam`

When you are finished with sam, enter a ^D or type:

`exit`

You will see:

`Connection closed.`

You are now back on the cluster server.

To administer file systems connected to an **auxiliary file server** (a cluster client that has a locally mounted file system) you must be logged in to the client in question. Details are in Chapter 8, "Introduction to Cluster Administration".

For More Information

- Chapter 11, "Reconfiguring the Kernel for a Cluster Node" provides rules and guidelines for reconfiguring a cluster client's kernel.
- Locally mounted file systems are described in detail in Chapter 12, "Adding Peripherals to a Cluster".
- There are tables showing where you should be logged in to perform various tasks near the beginning of Chapter 8, "Introduction to Cluster Administration", which provides a general introduction to administering a cluster.
- Before you begin administering a cluster, you should also read Chapter 9, "Backing Up Files in a Cluster", Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", and Chapter 13, "Managing Users in a Cluster".

What Is Context?

Context is the means by which particular cluster nodes are differentiated. (In fact, every HP-UX system, whether or not it is part of a cluster, has a context that is set at boot time.)

A computer's context is inherited by all **processes** (running programs) on that computer and is used in transactions between members of the cluster. For example, if a process on `client1` needs to open a **context-dependent file** (see below), the server determines the appropriate version of the file from the context of the requesting process. In this instance, the context will include the string `client1`.

A context consists of the following attributes:

cluster node name
: The cluster node name is the name you enter onto the SAM screen when you add the cluster node (or create the cluster). It is recorded in field 3 of the `/etc/clusterconf` file (see Chapter 8, "Introduction to Cluster Administration").

floating point hardware type
: This can be one or more of the following:

- `HP98248A` for floating point accelerator.
- `HP-MC68881` for Motorola coprocessor.
- `HP98635A` for floating point math card.

If a cluster node has more than one of these floating point hardware types, they will appear in the context string in the order listed.

If there is no floating point hardware on this machine, this field will not appear in the context string.

processor type
: Examples of processor types are:

- `HP-MC68040 HP-MC68020 HP-MC68010`
- `HP-MC68020 HP-MC68010`

cluster node type
: This can be either `localroot` or `remoteroot`. It is `localroot` if the root file system resides on the local machine (true for the cluster server or a standalone machine). It is `remoteroot` if the root file system is not on the local machine (true for cluster clients).

`default`
: All context strings end with the string `default`.

The command `getcontext(1)` shows the context string for the computer on which it's executed.

For example, if someone were to issue the `getcontext` command on system `client2` from Figure 2-1, they would see the following output:

```
client2 HP-MC98635A HP-MC68020 HP-MC68010 remoteroot default
```

Context-Dependent Files

Even though all cluster nodes in a cluster share the same file system, there are some files that must not be shared. Some examples of files whose contents should not or cannot be shared are device files and files that contain system setup scripts, such as `/etc/inittab`.

To allow files specific to a cluster node or class of cluster nodes, Hewlett-Packard has developed **context-dependent files**.

A context-dependent file (or CDF) is a mechanism for allowing different cluster nodes to see different contents for a file that has the same name on all cluster nodes. It is really a directory that looks like a file. (A directory can itself be a CDF, and in this case it's a nest of directories that looks like a single directory.)

The files in a context-dependent file are known as **elements**.

Context Attributes and CDF Elements

The name of each **element** in a context-dependent file must match the value of one of the **attributes** in the context string: cluster node name, floating point hardware types, processor type, cluster node type, or the string `default` (see the previous section, "What Is Context" for details).

As a rule, all the elements of a CDF should match the same kind of context attribute. For example, each of the elements of the CDF `/etc/inittab` corresponds to the name of a node in the cluster; that is, in each case a CDF element matches a *cluster node name* context attribute.

In Figure 2-2, the CDF `/etc/inittab` contains the elements `server`, `client1` and `client2`.

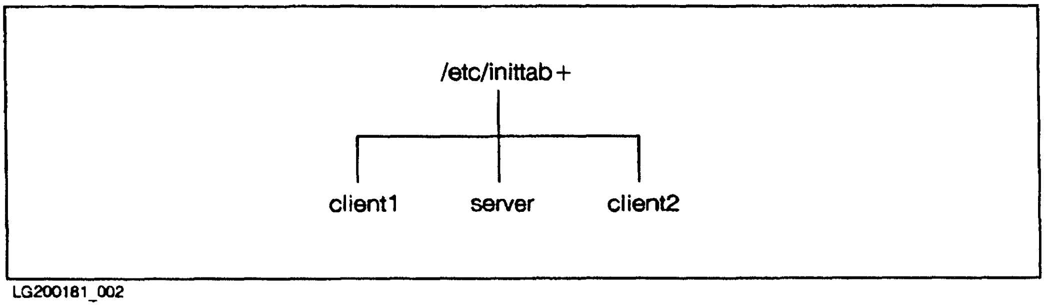


Figure 2-2. /etc/inittab in a Cluster

A user who types

```
more /etc/inittab
```

on system `client1` will see the element `/etc/inittab+/client1`, whereas a user on `client2` will see `/etc/inittab+/client2`, and the contents may be different. We call this kind of CDF a **node-specific** CDF, because the elements match the cluster node names.

System and Application Context-Dependent Files

Many HP-UX system files are automatically converted to CDFs when you configure an HP-UX cluster.

Caution Do not change the structure of system CDFs.

When you add application software to the cluster, you may need to create context-dependent files.

The remainder of this section should help you understand what CDFs are for and how they work. Chapter 8, "Introduction to Cluster Administration" provides more examples.

Some Reasons for Having Context-Dependent Files

- To reflect different configurations of peripherals.

  For instance, `/etc/inittab` and `/etc/ttytype` must show what terminals are configured on each cluster node. In our sample cluster, `server` might have just a console, `client1` might have a console plus three terminals, while `client2` might have a console plus one terminal. `Server`'s console might be of type 300l, while `client1`'s and `client2`'s consoles might be of type 300h.

- To report on specific cluster nodes.

  For example, `/etc/utmp` (read by the `who` command) must be a CDF so that you can see who is working on a given cluster node.

- To tune the kernel for each cluster node.

  When the SAM utility configures a cluster, it makes `/hp-ux` (the kernel) into a CDF. This allows you to tune the kernel and operating system parameters for each cluster node.

- To accommodate different floating point hardware.

  If you have different floating point hardware on different cluster nodes, and you need different versions of a program for each type of floating point hardware, you can create a CDF to hold the various versions.

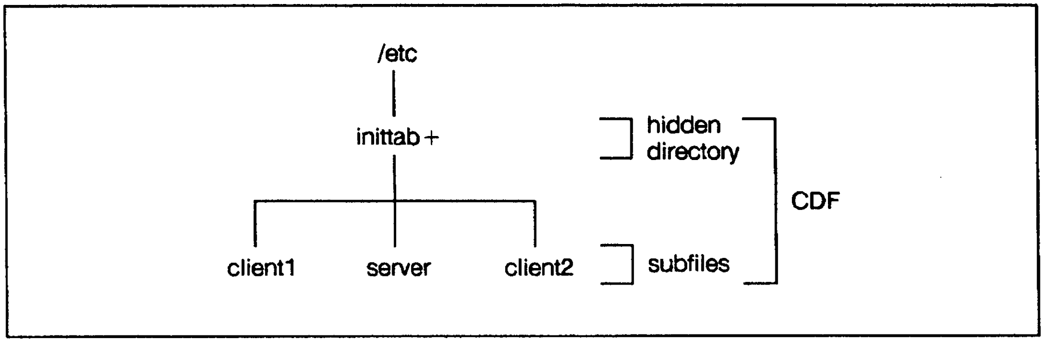
  There's more about this in Chapter 8, "Introduction to Cluster Administration".

How Context-Dependent Files Work

A context-dependent file is a special directory containing all the versions of a file needed by the different systems in the cluster.

For this reason a context-dependent file is also known as a **hidden directory**. It is called "hidden" because the directory structure is normally not seen: you think of it as a file, and normally see it as a file, but it is actually a directory.

For example, `/etc/inittab` looks something like this in a cluster.



LG200181_003

Figure 2-3. Structure of a Context-Dependent File

The "subfiles" in this special directory are known as **elements**. The elements can be any type of file, including another CDF or a directory.

The **element** name by which a given cluster node recognizes its "own" version must match an **attribute** in that cluster node's context. If no element matches any attribute in a cluster node's context string, it will appear to someone logged in on that cluster node that the file in question does not exist.

The + appended to the file name inittab in Figure 2-3 is not really part of the file name. It is an escape mechanism to allow you to see the elements of the CDF, and to refer to a particular element explicitly.

To look at all the elements in a CDF, append the + escape character to the file name, as in the examples that follow.

To list the elements of /etc/inittab, enter:

```
ll /etc/inittab+
```

In our sample cluster, you would see:

```
 server  client1    client2
```

You can also use the -H option to list the elements:

```
ll -H /etc/inittab
```

You can use the same mechanism to look at a specific element of a context-dependent file. For example, if you want to look at the contents of /etc/inittab that are specific to client1, and you don't happen to be logged in to client1, you can enter:

```
more /etc/inittab+/client1
```

This will show you client1's version of /etc/inittab, no matter which cluster node you are logged in to when you type the command. On the other hand, if you merely type

```
more /etc/inittab
```

you will see those contents of /etc/inittab which pertain to the node on which you are logged in.

How Context-Dependent Files Are Created

All the CDFs needed to run an HP-UX cluster are automatically created when you use the SAM program to create a cluster.

Caution Do not change a system file that is not a CDF to a CDF, or *vice versa*, and do not change the names of the elements.

Manage CDFs carefully and do not create them unless you really need to.

If you must make extensive use of CDFs, you might consider implementing ways to make them more visible; for example, by aliasing `ls` to `ls -H`. This will help make it clear when you are dealing with CDFs.

If you need to create your own CDFs, use the `makecdf` command. There are examples in Chapter 8, "Introduction to Cluster Administration", and the `makecdf(1M)` entry in the *HP-UX Reference* has more information.

Hewlett-Packard recommends naming elements of a given context-dependent file after only one type of context attribute: for example `localroot/remoteroot` only, or cluster node names only, or processor types only. Mixing context attributes can result in anomalous behavior, as Chapter 8, "Introduction to Cluster Administration", explains.

Autocreation

What happens when you move or copy a regular HP-UX file to a CDF depends on where you are logged in when you do it, and what elements are already in the CDF.

If the Element Already Exists

If you move or copy a regular HP-UX file to a CDF, and the CDF already has an element that matches an attribute in the context of the system to which you are logged in, then the contents of that element are overwritten.

For example, if a CDF named `/users/jme/cdffile` contains the element `/users/jme/cdffile+/client1`, and you execute the following command on system `client1`,

```
cp /tmp/flatfile /users/jme/cdffile
```

you will overwrite the contents of `/users/jme/cdffile+/client1` with whatever is in `/tmp/flatfile`.

If the Element Does Not Exist

If you move or copy a regular HP-UX file to a CDF, and the CDF does not yet have an element that matches any attribute in the system to which you are logged in, you will create a new element in the CDF.

By default this new element will be named after the cluster node you are logged in to.

For example, suppose the CDF `/users/jme/cdffile` contains only the elements `/users/jme/cdffile+/server` and `/users/jme/cdffile+/client2`.

If you execute the following command on system `client1`,

```
cp /tmp/flatfile /users/jme/cdffile
```

you will create the new element `/users/jme/cdffile+/client1`.

This element will contain whatever is in `/tmp/flatfile`.

How HP-UX Selects CDF Elements

When you access a context-dependent file such as `/etc/inittab` from a cluster node, HP-UX finds the correct version for the cluster node in question (that is, it finds the **element** of the CDF that matches a **context attribute** of the node you are logged in to).

It does this by scanning the elements of the context-dependent file in the same order as the attributes are listed by the `getcontext` command: it tries to match the node name first, then the floating point hardware type, if any, then the processor type, and so on (see "What Is Context", earlier in this chapter, for the full list).

HP-UX does this unless you use the + escape character to identify a CDF element explicitly.

Given the sample cluster shown in Figure 2-1, the following are examples of what will happen:

- If you specify `/etc/inittab` on the cluster server (whose node name in our example is `server`) you will see the element `/etc/inittab+/server`.
- If you specify `/etc/inittab` on the system `client2` you will see the element `/etc/inittab+/client2`.
- If you specify `/etc/inittab` on the system `client1` you will see the element `/etc/inittab+/client1`.
- On any system, if you specify `/etc/inittab+/client1` you will see the element named `/etc/inittab+/client1`.

Cluster Server Processes (CSPs)

Cluster Server Processes (CSPs) are kernel processes that handle requests from remote cluster nodes (or by certain local activities).

CSP functions include:

- All file system requests (e.g., opens, reads, writes, mount table updates).
- Swap space allocation requests.
- `sync` requests.
- PID (Process ID) allocation.

Requests that do not require CSPs include some network protocol messages and clock synchronization.

There are three types of Cluster Server Processes: Limited CSPs (LCSP), General CSPs (GCSP), and User CSPs (UCSP).

The limited and general CSPs are kernel processes. Even though these are processes that run in the kernel, they are shown by the `ps` command.

A User CSP is a special program that runs in user address space to perform some operation on behalf of the kernel.

Limited CSP

There is one LCSP on each cluster node. It is automatically spawned by the kernel at **cluster time** (on the cluster server) or at boot time (for cluster clients). **Cluster time** is when the cluster server executes the `cluster` command (from the `/etc/rc` file). The LCSP handles certain essential requests if no GCSP is available. It performs limited specific operations such as `syncs` and `mounts`.

An LCSP is the only type of CSP used by a cluster client that has no local disks. This one CSP is sufficient to handle incoming requests.

General CSP

The GCSPs are created when the `/etc/csp` command is executed (generally from `/etc/rc`). The number of GCSPs to create is given as an argument to the `/etc/csp` command.

The `/etc/csp` command with no argument will read the `/etc/clusterconf` entry for the cluster node to determine the number of GCSPs which should be running on that cluster node. (`/etc/clusterconf` is a file describing the cluster's configuration; it is described in detail in Chapter 8, "Introduction to Cluster Administration".)

The number of GCSPs is the last field of an `/etc/clusterconf` entry. In the default setup (created by the SAM program), the `/etc/csp` command in `/etc/rc` has no arguments.

SAM configures the cluster root server to have four GCSPs and the cluster clients to have one GCSP initially, and four if you convert the client to be an **auxiliary server** (see Chapter 12, "Adding Peripherals to a Cluster").

There should be a pool of GCSPs running at all times on the cluster server to handle remote requests from cluster clients.

If, for example, the `/etc/clusterconf` entry for your cluster server is:

```
08000900399d:1:server:r:1:4
```

the `/etc/csp` command run from `/etc/rc` will create four GCSPs.

If you now execute the command: `/etc/csp 5`, the system will create an additional CSP to bring the total up to five. The system may also create CSPs automatically if system load requires it.

If you execute the command `/etc/csp 0` on the cluster server, you will terminate all GCSPs on the cluster server. This will cause all cluster clients to stop functioning and ultimately crash. This is the only way to terminate GCSPs: you can't use the `kill` command.

Because the GCSPs will finish servicing all existing requests in progress, the effect of issuing the `/etc/csp` command may not be immediate.

The number of CSPs can never exceed the number specified by the ngcsp configurable operating system parameter. This parameter is originally set by SAM, but you may need to change it. For more information about ngcsp and its effect, refer to the ngcsp entry in the appendix on configurable operating system parameters in the *System Administration Tasks* manual.

See Chapter 12, "Adding Peripherals to a Cluster", under "Converting a Client to an Auxiliary Swap Server", for an example of a case where you need to change ngcsp.

User CSP

UCSPs service requests that cannot be handled in the kernel. They are created on demand when specific types of requests are received. The UCSP terminates after servicing the request.

Process IDs (PIDs)

Processes executing on different cluster nodes in a cluster must have unique Process IDs because many HP-UX programs use the PID in temporary filenames to guarantee unique file names.

In a cluster, the cluster root server allocates PIDs for the cluster clients. To save the network traffic that would be caused if a cluster client had to go to the root server for each PID, PIDs are allocated in "chunks" of 50 at a time.

Each cluster node keeps track of the PID chunks allocated to it and maintains a table of available PIDs. PIDs in this table are not recycled, but once a whole PID "chunk" is free, all 50 PIDs are returned to the server and may then be reallocated to another cluster node.

Certain PID numbers are reserved for kernel processes, and in these cases the same process will have the same PID number on every node in the cluster. For example, `/etc/init` always has a PID of 1.

Swapping

- The cluster root server must swap to its own disk space.
- A cluster client can swap to any one of the following:
  - To the cluster root server's swap area (remote or shared swap).
  - To the client's own local disk (local swap).
  - To another cluster client's local disk (distributed swap).

  Use the SAM program to configure where a client will swap. Detailed rules, guidelines and directions are in Chapter 12, "Adding Peripherals to a Cluster".

3

Setting Up the Network

Before you can create a cluster you must install and configure a **Local Area Network (LAN)** for it to run on. A Local Area Network is a group of computers communicating with each other via cable connections according to a standard protocol, such as IEEE 802.3 or Ethernet. The computers are normally in the same building or on the same site, hence "local".

This chapter provides a summary of what you need to do to set up the network that will form the basis of your cluster, and points you to the networking documentation you will need to go to for details. This chapter also contains rules and guidelines for setting up a network to support a cluster.

Outline of Steps To Set Up the Network for a Cluster.

Note This is intended only as a quick reference, to help you plan the task. You must read the documentation mentioned at each step before attempting to perform the step.

1. Familiarize yourself with networking terms. See *Installing and Administering LAN/9000 Software*.
2. Install and test the LAN hardware on each computer that is to be a member of the cluster (the cluster server and cluster clients). Use the following manuals:

   HP 98643A LAN/300 Link LANIC Installation Manual

   LAN Cable and Accessories Installation Manual
3. Design the network.
   a. Read the next section in this chapter, "Networking Rules for an HP-UX Cluster".
   b. Create a network map.

      This should show the **network nodes** (the computers in the network), their physical locations, the connections between them, their network names and addresses and other information detailed in *Installing and Administering LAN/9000 Software*.

c. Decide whether your cluster will be an isolated network, or will be connected to another network or other networks.

   If it will be connected to another network, read the section "Recommendations" in Chapter 6, "Connecting the Cluster to Another Network".

4. Install and configure the networking software.

   See *Installing and Updating HP-UX*, Series 300/400 version, for guidance on installing HP software via the `update(1M)` utility.

   a. Install the LAN software onto the computer that is to be the **cluster root server** (the one that will have root file system disk attached to it).

      See *Installing and Administering LAN/9000 Software.*

   b. Install the ARPA services software onto the cluster root server.

   c. Configure the ARPA software.

      See *Installing and Administering ARPA Services.*

You are now ready to begin setting up your cluster. Directions are in Chapter 4, "Setting Up a Cluster".

Networking Rules for an HP-UX Cluster

- All cluster nodes (all members of the cluster) must be on a single LAN. This means:
  - Cluster nodes cannot communicate *with each other* via a gateway, (but a cluster can communicate *with other networks* via a gateway).
  - Cluster nodes *can* communicate with each other over bridges and repeaters.

In addition, we recommend the following:

- Dedicate a LAN to the cluster.
- Use the cluster root server, rather than a cluster client, as the gateway to other networks.

  See Chapter 6, "Connecting the Cluster to Another Network".

Networking Names

Networking services use the following four types of name to identify network nodes (individual machines within the network):

uname The node's SVID/SVVS (System V Interface Definition and System V Validation Suite) name.

Must be 8 characters or less.

host name The node's ARPA host name.

Consists of a *name* portion plus *domain* extensions.

node name The node's NS node name (if you are using NS services).

cname The node's cluster node name (In this book, we will normally spell out **cluster node name**).

To make everything work correctly, **uname**, the *name* portion of **host name** and **cname** (cluster node name) must all be the same. If you are using NS services, **node name** should preferably be the same as well, but it's not essential.

Note Because of this, the name you pick must be 8 characters or less, since 8 characters is the maximum allowed for **uname** by SVID/SVVS, even though the theoretical maximum for each of the other names is greater.

When you configure the cluster server, SAM will verify that these names match as they should.

4

Setting up a Cluster

This chapter explains how to configure a Series 300 or 400 computer as a cluster root server, the first step in creating a cluster. A Series 300 or 400 computer can be the root server for Series 300 and 400 clients only. If your cluster will contain any Series 700 computers, then the root server must also be a Series 700, and you will need a different version of this manual, part number B2355-90038.

Before you continue you should have a good idea of what a cluster is (see chapters 1 and 2), and you should have installed and configured the network (see chapter 3).

What You're Going To Do (Summary)

Configuring a cluster server requires the following steps (references are to later sections of this chapter):

1. Be sure that you have all the necessary hardware and software.

   See "What You Need Before You Start".

2. Gather and write down LAN and ARPA information about the computers that will comprise the cluster.

   See "Gathering Information".

3. Configure the cluster server, using the SAM utility.

   See "Configuring the Cluster Server".

You are then ready to start adding cluster clients. That procedure is explained in chapter 5.

What You Need Before You Start

This section outlines the hardware and software requirements for building a cluster.

Note These are guidelines. Satisfying minimum requirements may not be enough, depending on how you configure the cluster and how it's used. For example, a memory-intensive application such as X-Windows will probably not run well on a Series 300/400 cluster client that has less than 8 megabytes of memory.

Cluster root server:

HP 9000 Series 300/400 computers with a model number higher than 350 can be servers for a cluster consisting of other Series 300s and 400s, subject to the hardware restrictions below.

The server must have:

- At least 16 Mbytes of RAM (24-32 Mbytes recommended).
- At least one disk drive, of at least 400 Mbyte capacity.
- The following products installed:
  - □ Release 9.0 or later of HP-UX.
  - □ LAN/9000 (conforming to 802.3 standard).
  - □ NS/9000.
  - □ ARPA Services/9000.

Cluster clients:

No Series 800 computer can be a client of a Series 300/400 root server or vice versa.

- Series 300 computers with a model number higher than 350 can be cluster clients, if they fulfill the hardware requirements listed below.
- Any Series 400 computer can be a cluster client, subject to the requirements below.

A Series 300/400 cluster client must have:

- Rev B or later Boot ROM

  (If you're not sure what version of the Boot ROM you have, you can check by booting in **attended mode**. See Chapter 5, "Adding Cluster Clients", under "What You Need Before You Start", for more information.)
- At least 3 Mbytes of RAM
- LAN hardware

Gathering Information

Now you need to gather information about the computers that will be in your cluster. Specifically, you need the following for the root server and each cluster client:

- ARPA host name
- Internet Address
- LAN card station address (link level address)

The following pages contain explanations of each of these items, followed by some blank forms to help you organize the information. The first few lines of the first form are filled out for a sample cluster.

Use the explanations and the example that follows them to help you fill out a form for your cluster.

Note Be sure to read the explanations that follow before you start filling out the form. In some cases the information you need will be on the SAM screen when you go to configure the cluster server or add the client; in others you'll need to do some research. The explanations make clear what's necessary in each case.

ARPA Host Name

This name must be no more than eight characters long and must be unique within the cluster. See Chapter 3, "Setting Up the Network", under "Networking Names".

Cluster root server. You can leave this blank for the root server if the server already has a host name: SAM (the system administration tool you will be using to configure the server) will get the name from the system (via `uname(2)`).

Cluster client. See "Determining Host Name and Internet Address for a Client", a little later in this chapter.

Internet Address

Cluster server. You can leave the internet address for the root server blank on the worksheet if the server already has an internet address: SAM will find the address for you when you get to the `Create Cluster` screen (see "Using SAM to Configure the Cluster Server", later in this chapter).

Cluster client. See "Determining Host Name and Internet Address for a Client", a little later in this chapter.

What is an Internet Address?

The **internet address** consists of a **network address**, identifying a given network, and a **host address**, identifying a particular member of the network (or **network node**). The root server and all the cluster clients will have the same network address, but each will have its own host address.

Unique network addresses make it possible for a network to communicate with other networks around the world. If your network has not been assigned a unique network address, you can obtain a **Class C** internet address (see the next section for more information about Class C addresses) by contacting:

Government Systems Inc.
Attn.: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly
VA 22021
(800) 365-3642
(703) 802-4355

If you use a network address not assigned by the Network Information Center and you then need to connect the cluster LAN to other networks, you may need to change all the addresses in your network (for example, if there is an address conflict with another network).

Internet addresses are usually represented in the form

n.n.n.n

where *n* is a number from 0 to 255, inclusive. (This is referred to as **decimal dot** notation.) For example:

```
192.6.2.9
```

Note This is an example of a **Class C** address. It is likely, but not certain, that your address will be a Class C address. See *Installing and Administering LAN/9000 Software* for an explanation of the three classes of internet address.

In the above Class C example, the first three numbers (`192.6.2`) are the **network address**. The last number (`9`) is the **host address**.

Once you have a network address for the cluster, assign a unique host address to each of the cluster nodes.

Caution Do not use leading zeros in the components of the Internet address: a leading zero indicates an octal number, not a decimal number. (SAM will not allow leading zeros.)

In a Class C address, you can assign any number from 1 to 254, inclusive, as the host address.

Caution Do NOT assign 0 or 255 as host addresses in a Class C address. These are reserved addresses. Refer to *Installing and Administering LAN/9000 Software* for a full discussion of the rules governing internet addresses.

Determining Host Name and Internet Address for a Client

Note This section will help you fill out the cluster client entries on the worksheet that comes a little later in this chapter, but you're not actually going to use these entries for a while—until you get to the `Add Cluster Clients` screen in the SAM utility (covered in Chapter 5, "Adding Cluster Clients", in this manual). You may want to put a marker here so you can refer back if you need to when you get to the `Add Cluster Clients` screen.

When you add a cluster client in SAM (as described in chapter 5) you will first enter the client's **cluster node name**, the name by which it will be uniquely known in the cluster.

SAM then checks the `/etc/hosts` file (or the Domain Name Server, or NIS, whichever you configured the root server to use. See the networking documentation listed in Chapter 3, "Setting Up the Network", for more information.) If SAM finds the cluster node name you have entered, SAM will enter the corresponding internet address for you on the screen.

Be careful that the cluster node name you choose is not already in use by some other system in the same hosts file or internet domain. If SAM returns an internet address with a network portion that is not the same as the network portion of the root server's address, you should probably not use this name. (See "Internet Address", earlier in this chapter, to see what an internet address looks like.)

How you assign a cluster client's node name (which is also its network host name), and its internet address, depends on how the computer to be made a client was being used before. There are several possible cases, each of which is discussed below.

You'll find more information, if you need it, under "Internet Address", earlier in this chapter; and in Chapter 3, "Setting Up the Network" (which also supplies the names of networking documents you may need to consult if you have not already done so).

New System. In this case, assign the client a name and internet address that are not already being used in your system's `/etc/hosts` file (or Domain Name Server or NIS Server, whichever your cluster root server uses).

If the cluster is on its own LAN, as recommended in Chapter 3, "Setting Up the Network", this is easy: you just need to give each client its own name, and make sure that the network portion of the internet address is the same as the server's, and that the host portion is unique (see "Internet Address" earlier in this chapter). Remember that the name must be no more than eight characters.

Write the name and address on your worksheet.

System That Was Not Part of Any Network. Treat this client just as though it were a new system (see "New System" above).

System That Was Part of a Network. There are several possibilities here.

- If the system that is to be made a cluster client was already on the network the cluster will use, then:
  - If you will use the same name for the cluster node name as you were already using for this computer's network host name, simply write that name on your worksheet and leave the internet address blank: SAM will get it for you when you enter the name on the `Add Cluster Clients` screen.

    Remember that the name must be no more than eight characters.

    For example, suppose the system that you are going to convert to a cluster client has been functioning as a network node on the LAN that the cluster will use, and its name in `/etc/hosts` is `freddy`. If you are going to continue to use `freddy` as the cluster node name, just write `freddy` on the worksheet. You will enter `freddy` as the `Client Name` on the SAM `Add Cluster Clients` screen when you get there (see Chapter 5, "Adding Cluster Clients"). SAM will then fill in `freddy`'s internet address for you.

- If you will use a different name, then you must change the name in /etc/hosts (or on your Domain Name Server or NIS Server, whichever you have configured the cluster root server to use), write the new name on your worksheet and enter it onto the SAM Add Cluster Clients screen when you get there.

  This will ensure that the cluster client's host name and cluster node name match, as they must. You can leave the internet address field on your worksheet blank: SAM will get it for you when you enter the name on the Add Cluster Clients screen.

  Remember that the name must be no more than eight characters.

  For example, suppose the system that you are going to convert to a cluster client has been functioning as a network node on the LAN that the cluster will use, and its name in /etc/hosts is freddy. If you are going to name the cluster client client1, then you need to change freddy to client1 in /etc/hosts. Then write the name client1 on your worksheet.

  You will enter client1 as the Client Name on the SAM Add Cluster Clients screen when you get there (see Chapter 5, "Adding Cluster Clients"); SAM will then fill in client1's (formerly freddy's) internet address for you.

- If the system that is to be made a cluster client was on a network other than the one the cluster will use, then:
  - ◻ If you will use the same name for the cluster node name as you were already using for this computer's network host name, then you will need to "move" that system to your network. To do this, change the internet address in `/etc/hosts` (or on your Domain Name Server or NIS Server, whichever you configured the root server to use).

    The network portion of the address must match the network portion of the root server's internet address, and the host portion must be unique within the group of computers that share the common network address. (See "Internet Address", earlier in this chapter, for an explanation of the parts of an internet address.)

    Write the name on your worksheet. Remember that it must be no more than eight characters long. You can leave the internet address blank on the worksheet: SAM will get it for you when you enter the name on the `Add Cluster Clients` screen.

    For example, suppose the system that you are going to convert to a cluster client has been functioning as a network node on some other LAN than the one the cluster will use, and its name in `/etc/hosts` is `freddy`. If you are going to continue to use `freddy` as the cluster node name, you must change `freddy`'s internet address in `/etc/hosts` to point to the cluster LAN. Then write `freddy` on your worksheet.

    You will enter `freddy` as the `Client Name` on the SAM `Add Cluster Clients` screen when you get there (see Chapter 5, "Adding Cluster Clients"); SAM will then fill in `freddy`'s internet address for you.
  - ◻ If you will use a different name, then for housecleaning purposes you should probably delete this system's entry from `/etc/hosts` (or Domain Name Server or NIS Server).

    Then proceed as though this were a new system. See "New System" above.

Station (Link Level) Address

The last item on the information sheet is the LAN card's **station address**, also known as the **link level address**. Fill this in or leave it blank as directed below:

For All Cluster Clients

Leave this blank on your worksheet for now. You'll fill it in when you're ready to add the clients to the cluster (see Chapter 5, "Adding Cluster Clients").

For a Server That Has Only One LAN card

4

Leave this blank on your worksheet. SAM will get the number for you when you configure the server. (SAM refers to this address as the **link level address**. See "Using SAM to Configure the Server", later in this chapter).

For a Server That Has More Than One LAN card

You will be able to get a list of the LAN cards on your system, with their station (link level) addresses, when you get to the `Link Level Address` field of the `Create an HP-UX Cluster` screen in SAM.

If you want to see this information before you get to the SAM screen, you can use the `landiag` utility:

1. Enter

   `landiag`

2. The program displays a list of commands. Enter

   `l` *(the letter l, for lan)*

3. The program displays another list of commands. Enter

   `d` *(for display)*

4. The program displays information for **`Device file`**, **`Select code`**, **`current state`** and **`LAN Interface address, hex`**. The **`LAN Interface address`** is the link level address. Write down everything except the leading `0x`. For example, if this is what you see,

   ```
   LAN Interface address, hex     = 0x08000903F637
   ```

   then your link level address is `08000903F637`

Which LAN card?

Knowing the link level addresses may not be enough if you don't know which LAN card is the one attaching the server to the cluster.

The following guidelines should help:

- The default select code for a LAN card supplied with the system is 21, so if you have just installed a second LAN card for the cluster, the card you want is probably the one whose select code is *not* 21.
- If you're still not sure, turn off power to your machine, trace the cluster's cable back to the LAN card, and read the link level address off the card.

  You may have to lift the card out of the computer. This is a delicate operation; consult the documentation that came with the card, which should also tell you how to find the number on the card.

Write the link level address of the cluster's LAN card on the cluster worksheet.

Cluster Worksheet

Table 4-1. Sample Cluster Worksheet

Workstation	ARPA Host Name	Internet Address	Cluster LAN card's Station (Link Level) Address
root server	server	192.25.204.1	0x080009004a11
client #1	client1	192.25.204.2	0x0800090044ff
client #2	client2	192.25.204.3	0x08000900a63f
client #3	client3	192.25.204.4	0x08000902087a
client #4			
client #5			
client #6			
client #7			
client #8			
client #9			
client #10			

4

Cluster Worksheet

Workstation	ARPA Host Name	Internet Address	Cluster LAN card's Station (Link Level) Address
server			
client #1			
client #2			
client #3			
client #4			
client #5			
client #6			
client #7			
client #8			
client #9			
client #10			
client #11			
client #12			
client #13			
client #14			
client #15			
client #16			
client #17			
client #18			
client #19			
client #20			

Cluster Worksheet

Workstation	ARPA Host Name	Internet Address	Cluster LAN card's Station (Link Level) Address
client #21			
client #22			
client #23			
client #24			
client #25			
client #26			
client #27			
client #28			
client #29			
client #30			
client #31			
client #32			
client #33			
client #34			
client #35			
client #36			
client #37			
client #38			
client #39			
client #40			

Configuring the Cluster Server

Configuring a cluster root server means converting a standalone system to be a server for a cluster. It involves changes to the kernel and substantial changes to the file system, and it is irreversible, in that there is no automated way to "unconfigure" a server once it's been configured. Be quite sure that you want this system to be a cluster root server before you embark on the conversion.

Configuring the system as a server involves rebooting the system, so if the computer is already in use as a multi-user system, you will probably want to plan to do the configuration at a time when the fewest possible users will be inconvenienced.

Using SAM to Configure the Server

(If you have not used SAM before, or need assistance using SAM, you'll find a guide in Chapter 1, "Introduction to System Administration" in the *System Administration Tasks* manual).

Caution Once you have configured a system as a root server, there is no automated way to undo the substantial changes that SAM makes to the kernel and file system (see "What SAM Does For You", at the end of this chapter, for details of the changes).

4

1. Log in under a superuser ID.
2. Put your system in single-user mode by entering the command:

   `/etc/shutdown`

3. Wait for the system to make the transition into single-user mode (you'll see a shell prompt).
4. Run SAM
5. Select:

   `Cluster Configuration`

   SAM will ask you to confirm that you want create a cluster, and will warn you if you are not in single-user mode and ask you if you want to continue. If you did not shut down the system as described above, don't continue. Exit SAM and return to step 2.

   Now SAM will check whether this computer meets requirements for a root server. This takes a minute or so.

In case of error

Problem	What to do
Message about missing hardware/software	Read Help screen for details.
	Exit SAM.
	Install and configure missing hardware/software.
	Start again.
Message that you're on a cluster client	Read Help screen.
	If you really want this to become a root server, you must add and configure disk(s) for the file system, then reboot standalone, then configure the system as a root server.
Message about missing files	Look in `/tmp/cluster.log` for the names of the missing files and the filesets they belong to.
	Use the `update(1M)` utility to load the filesets. See Chapter 5, "Updating HP-UX", in the Series 300/400 version of *Installing and Updating HP-UX*.

You'll notice that the `Create Cluster` screen allows you to enter information about cluster clients as well as the server. Unless you're already familiar with the process of setting up a cluster, just fill in the information that applies to the server and use the procedure in chapter 5 for adding cluster clients.

6. Accept the default `Server Nodename` or enter it from your worksheet.

   If there's a value in this field, you can't change it.

   If you are prompted for a host name, type the root server's ARPA host name from your cluster information sheet.

   The node name can be one to eight ASCII characters; must not be `DEFAULT`, `default`, `localroo`, `remotero`, `UNKNOWN`, `unknown`, `HP-PA` or begin with `HP-MC`; and must not contain spaces, newline characters, or pound signs (`#`).

Note If you have two LAN cards (one of which will communicate with systems outside the cluster) both should use the same ARPA host name. See Chapter 6, "Connecting the Cluster to Another Network".

---

7. Accept the default `Machine Type`.

   This shows which type of computer the server is.

8. Accept the default `Internet Address` or enter it from your worksheet.

   There's probably an address already displayed in this field: it is the internet address associated with the cluster node name shown on the screen.

   If there's nothing displayed here, type the root server's internet address from your cluster information sheet.

9. Accept the default `Link Level Address` or select the one that matches what's on your worksheet.

   This is the LAN card's station (link level) address. This field is always filled in, but if the cluster server has more than one LAN card, the number displayed may not be the one you need.

   - If you have only one LAN card (the built-in LAN interface), the number already filled in is correct. You can't change it.
   - If you have more than one LAN card, check the number on the screen carefully against what you have on your cluster worksheet. If they don't match, click on the field and select the number that matches what you have your worksheet.

   If you don't know which is the right address, and you did not write it on your cluster information sheet, you need to go back to the section "Station (Link Level) Address" earlier in this chapter.

10. Press (OK).
11. You'll see a warning that it's difficult to convert the cluster server back to a standalone machine once the cluster configuration has been done.

    Assuming you do want to configure this computer as a root server, respond (Yes) to continue.

12. It takes several minutes for SAM to convert the system to a root server. You'll see messages telling you what's happening. See "What SAM Does For You", at the end of this chapter, for details.

    Once the configuration is complete, you'll be asked if you want SAM to reboot the system. Respond (Yes): the changes will not take effect until you have rebooted the system.

What To Do Next

You are nearly ready to add cluster clients. There are a couple of housekeeping tasks to do first:

1. Save a hard copy of the file `/tmp/cluster.log`, which contains a list of the **context-dependent files (CDFs)** which SAM has created on your system.

   A context-dependent file is a file whose contents differ depending on which member of the cluster is using it: see Chapter 2, "Understanding Clusters", for a full explanation.

2. Read the next section, "What SAM Does For You".

   Then turn to Chapter 5, "Adding Cluster Clients".

What SAM Does For You

In configuring the cluster root server, SAM does the following:

- Creates the `/etc/clusterconf` file.

  There will be two lines in the `/etc/clusterconf` file at this point. The first is a special entry that contains the server's station address. The second line is a standard entry, defining the server.

  There's a discussion of the format of entries in `/etc/clusterconf` in Chapter 8, "Introduction to Cluster Administration", in this manual, and the file is also discussed under `clusterconf(4)` in the *HP-UX Reference*.

- Turns certain files into CDFs

  For a list of the files that are now CDFs, refer to the file `/tmp/cluster.log`. If you have not already printed this file, do so now and save it for future reference.

- Modifies the kernel to include cluster configuration.

  The new kernel is built from the existing compiled kernel (`/hp-ux`), which has now been converted to a context-dependent file. (See Chapter 2, "Understanding Clusters", for an explanation of context-dependent files.)

  The full pathname for the cluster server's version of the kernel is now `/hp-ux+/`*server_nodename*. For example, if the server's cluster node name is `server`, the kernel will be in `/hp-ux+/server`.

  This kernel has the drivers and the parameter values needed by a cluster server.

  The old kernel has been saved as `/SYSBCKUP` and also copied to `/hp-ux+/standalone`).

  The file `/etc/conf/dfile+/`*server_nodename* contains the source text that matches the kernel. For example, if the server's cluster node name is `server`, the `dfile` that matches the compiled kernel will be in `/etc/conf/dfile+/server`.

  (While logged in to the server, you can access this file simply as `/etc/conf/dfile`.)

- Checks that the LAN device file is present and of the right type, and warns you if it's not.
- Puts an entry for the root server in each of the following files (unless the entry was already there):
  - □ `/etc/hosts` (or in "hosts" database)
  - □ `/etc/hosts.equiv`
  - □ `$HOME/.rhosts` (root's home directory)
  - □ `/etc/X0.hosts` (if it exists)

5

Adding Cluster Clients

This chapter explains how to add clients to a cluster.

Before you do any of the tasks described in this chapter, you need to configure the cluster server. Chapter 4, "Setting Up a Cluster", explains how to do this.

5

Distributing Disks in a Cluster

Before you add cluster clients, you should think about how you want to distribute the cluster's peripherals.

Your options for each class of peripheral are spelled out in Chapter 12, "Adding Peripherals to a Cluster", in this manual. The most important, and the most complex, options concern disk drives.

HP-UX gives you considerable (though not unlimited) flexibility in distributing file space and swap space in a cluster. All of the following configurations are permitted:

- All file system space is on the server's disks and all clients swap to the server's disk space. (All the cluster's disks are attached to the server.)
- All file system space is on the server's disks, some clients swap to the server's disks and some swap to their own disks.
- All file system space is on the server's disks, some clients swap to the server's disks, some swap to their own disks, and some swap to another client's disks.
- File system space is distributed between server and clients, but all clients swap to the server's disk space.
- Both file system and swap space are distributed between the server and clients.

File systems residing on a client's disks are referred to as **locally mounted file systems**. Swap space on a client's disk is usually called **local swap**. There are restrictions governing both file system space and swap. These are spelled out in Chapter 12, "Adding Peripherals to a Cluster", which also contains a discussion intended to help you decide what configuration will suit you best.

What You're Going To Do (Summary)

Adding cluster clients requires the following steps:

1. Get each client's station address (link level address) and prepare the client to boot over the LAN *(on client)*
2. Use SAM to add the cluster clients *(on server)*
3. Boot each client *(on client, after all clients have been added)*
4. Optionally add local disks.

What You Need Before You Start

Before adding cluster clients make sure that all of the following are true:

- The cluster server is configured.

  See Chapter 4, "Setting Up a Cluster".

  - If any cluster client is a Series 700 computer, the server must be a Series 700 computer.

    In this case, you need a different version of this manual, part number B2355-90038.

- Each computer to be made a cluster client is supported as such.

  Guidelines are in Chapter 4, "Setting Up a Cluster". Consult your Hewlett Packard SR or SE if you are in doubt about a particular configuration.

- Each Series 300/400 computer which is to be made a cluster client has Rev. B or later boot ROM.

  The "Rev." (revision) letter prints on the screen when you boot the computer. It is one of the first few lines in the display and scrolls off the screen quickly. If you boot in **attended mode**, you will halt the display while the Boot ROM revision is still showing. The section "Preparing To Add a Series 300/400 Client", later in this chapter, explains how to boot in attended mode. The line you're looking for will say something like

  ```
  BOOTROM Rev. C
  ```

- Each computer to be made a cluster client has at least 3 Mbytes of RAM.
- You have filled out a worksheet for the cluster, following directions in Chapter 4, "Setting Up a Cluster".

Preparing To Add a Series 300/400 Client

The computers in your cluster will communicate with each other over your LAN. In order to do that they will need to know each other's LAN **station address** (also known as a **link level address**).

Before you can add a Series 300/400 client to the cluster, you will need to get its LAN card's station address. The following section explains how.

Complete the steps in this section for each Series 300/400 computer to be made a cluster client, then go on to "Using SAM To Add Cluster Clients".

Getting the Station (Link Level) Address

Perform this procedure on the computer you are going to add as a cluster client, *not on the cluster server.*

5

You will need the cluster worksheet you started when you configured the cluster server. (See Chapter 4, "Setting Up a Cluster".)

If the Series 300/400 Workstation Is Currently Running

If the Series 300/400 workstation that you want to use as a client in your cluster is currently running (in standalone mode or as a member of another cluster), you can get the address of its LAN card using the utility `/usr/bin/landiag`. To do this:

1. Log in to the Series 300/400 workstation.
2. Run `/usr/bin/landiag`

   You will see `landiag`'s main menu, which will give you the following choices:

```
lan      = LAN Interface Diagnostic
menu     = Display this menu
quit     = Terminate the Diagnostic
remote   = Remote Node Communications Diagnostic
terse    = Do not display command menu
verbose  = Display command menu
```

3. Enter the `lan` command:

   ```
   Enter command:lan
   ```

   `Landiag` will then display the `lan` submenu, which looks like this:

   ```
   clear    = Clear statistics registers
   display  = Display LAN Interface status and statistics registers
   end      = End LAN Interface Diagnostic, return to Test Selection
   menu     = Display this menu
   name     = Name of the LAN Interface device file
   quit     = Terminate the Diagnostic, return to shell
   reset    = Reset LAN Interface to execute its selftest
   ```

4. Enter the `display` command:

   ```
   Enter command:display
   ```

   `Landiag` will then display the status of your built-in LAN card. The first part of the display will look similar to this:

   ```
                                LAN INTERFACE STATUS DISPLAY
                                 Mon,Apr 22,1992     15:53:46

   Device file                        = /dev/lan
   Select code                        = 0
   Current state                      = active
   LAN Interface address, hex         = 0x080009nnnnnn
   Number of multicast addresses      = 4
   Frames received                    = 8022922
                                  ⋮
   ```

   The station (link level) address is listed as "`LAN Interface address, hex`" (highlighted in the above example).

5. Write down the station address for this workstation on your worksheet from Chapter 4, "Setting Up a Cluster". You will use it later, when you run SAM.

6. To exit the `landiag` utility, press (Return) and then enter the `quit` command:

   ```
   Enter command:quit
   ```

If the Series 300/400 Workstation Is Not Currently Running

You will need the cluster worksheet you started when you configured the cluster server. (See Chapter 4, "Setting Up a Cluster".)

Procedure

Do the following steps on the workstation you are going to add as a cluster client, *not on the cluster server.*

1. If any disk attached to the *cluster client computer* has a bootable system on it, it's a good idea to turn off the disk (because you don't want to boot from that disk).

   If you decide to leave the disk attached and powered on with a bootable system on it (if this client is going to double as a standalone system), you will need to be careful to pick the right operating system when you go to boot the client over the LAN: the system you need will appear under the `LAN` statement in the upper right corner of your screen. "Booting The New Client (Series 300/400)", later in this chapter, gives details.
2. Turn on power to the computer.
3. Press the space bar and continue to hold it down until you see the word `keyboard` on the left of the console screen.

5

You will see a screen similar to the one shown in Figure 5-1.

```
Copyright 1987,
Hewlett-Packard Company.
All Rights Reserved.

BOOTROM Rev. C
Bit Mapped Display
MC68050 Processor
Keyboard
HP-IB
HP98620B
HP98644 at 9
HP98625 at 14
HP98643 at 21, 080009000001
4182016 Bytes




SEARCHING FOR A SYSTEM (RETURN To Pause)
RESET To Power-Up
```

Figure 5-1. Sample Boot ROM Display

A LAN card is declared as follows:

lancard-part-number `at` *select-code*`,` *station-address*

(for example, `HP98643 at 21, 080009000001`).

4. Write down the cluster LAN card's station (link level) address on the cluster worksheet.
5. Leave the cluster client in this state. (This is called **attended boot mode.**) You will come back later to finish the boot sequence.

Complete the above steps for each Series 300/400 computer to be made a cluster client, then go on to "Using SAM To Add Cluster Clients".

5

Using SAM To Add Cluster Clients

Before you can start adding clients:

1. Run SAM *on the cluster server*.
2. Select:

   `Cluster Configuration ->`

   Then choose

   `Add Cluster Clients ...`

   from the `Actions` menu.

In case of error

Problem	What to do
Message that this is not a cluster server	You have not configured this computer as a cluster server. Check that you are on the cluster server. If you have not yet configured a cluster server, do so now. Directions are in Chapter 4, "Setting Up a Cluster".

For each cluster client, fill out the fields on this screen as follows:

1. Type the `Client Name`.

   This is the ARPA host name for this cluster client from your cluster worksheet.

   The name must be at least one and no more than eight characters (any ASCII characters) and must be unique to the network.

   The name cannot be any of the following:

   ```
   DEFAULT
   default
   localroo
   remotero
   UNKNOWN
   unknown
   HP-PA
   ```

   and cannot be anything beginning with the characters `HP-MC`.

   See Chapter 3, "Setting Up the Network", under "Networking Names", and Chapter 4, "Setting Up a Cluster", under "Determining Host Name and Internet Address for a Client", for more information.

2. Accept the default `Machine Type: 300/400`.
3. Enter the cluster client's ARPA internet address or accept the default.

   If SAM supplies an address here, it is the address associated with the ARPA host name you entered under `Client Name`.

   See "Determining Host Name and Internet Address for a Client" (in Chapter 4, "Setting Up a Cluster"), for more information.

4. Enter the cluster client's LAN interface station address (link level address) into the `Link Level Address` field.

   You wrote this number down on your cluster worksheet. (If you didn't, and you don't know the address, go back to the section on getting the station address earlier in this chapter.)

5. Press (Add), then enter the information for the next cluster client, if you have more clients to add.

When you have added the information for all the cluster clients, do the following:

1. Press (OK).
2. Wait for confirmation that the clients were added.
3. Check `/tmp/cluster.log` to see if any errors occurred while you were adding clients.
4. Boot each new client.

   The section that follows, "Booting The New Client (Series 300/400)", explains what to do.
5. Add local disk drives if you need to.

   See "Adding a Local Disk" later in this chapter.

Booting The New Client (Series 300/400)

Do this after you have added all your cluster clients via SAM.

For a Client That Was Previously Running

1. Reboot the client.

   - If the client was previously running as a standalone system:

     - If you have removed or turned off the disk drives attached to the client, simply reboot the client. It will boot over the LAN.

     - If there is a bootable system on any disk attached to the client, do the following:

       a. Reboot the client in **attended mode** (as described earlier in this chapter, under "If the Series 300/400 Workstation Is Not Currently Running").

       b. From the operating system entries on the right hand side of the screen, choose the `SYSHPUX` entry under the `LAN` ... statement that identifies the cluster server. (See the next section, "For a Client That Was Not Previously Running", if you need more help.)

   - If the client was previously part of another cluster:

     - If you have removed the client from the other cluster, simply reboot the client. It will boot over the LAN.

       (See Chapter 7, "Removing and Renaming Cluster Clients", for instructions on removing a client from a cluster.)

     - If the client is still a member of another cluster (if it's still recorded in another cluster server's `/etc/clusterconf`), do the following:

       a. Reboot the client in **attended mode** (as described earlier in this chapter, under "If the Series 300/400 Worksation Is Not Currently Running").

       b. From the operating system entries on the right hand side of the screen, choose the `SYSHPUX` entry under the `LAN` ... statement that identifies this cluster's server. (See the next section, "For a Client That Was Not Previously Running", if you need more help.)

2. When the cluster client has booted, you will see a login prompt. Log in.

   Remember that you need to use a user ID and password that are valid on the cluster root server.

   If the login fails, you may be booted to the wrong system: follow the troubleshooting directions in the section called "Troubleshooting Boot Problems", later in this chapter.

3. Now boot the next client, and then proceed to "Adding a Local Disk" if you need to.

For a Client That Was Not Previously Running

1. *Return to the cluster client*, which you left in attended boot mode.

   The console should have a set of entries in the upper right corner looking something like this:

   ```
   LAN, 21, server   (the name of the cluster server)
    1H SYSHPUX
    1D SYSDEBUG
    1B SYSBCKUP
   ```

2. Choose the SYSHPUX option that appears under the LAN, ... statement by typing that option's reference number.

   In this example you'd type 1H (or 1H (Return) if this is a Rev. D Boot ROM). 5

In case of error

Problem	What to do
Boot fails	Check /tmp/cluster.log (on the server) to see if any errors occurred while you were using SAM to add the client. Check that rbootd is running on the server, and check /usr/adm/rbootd.log. Check the LAN cable connections from your client to the server. Reboot the client (turn the power off and then on).
Client already booted, or will not boot over LAN	See "Troubleshooting Boot Problems" below.

3. When the cluster client has booted, you will see a login prompt. Log in.

   Remember that you need to use a user ID and password that are valid on the server.

   If the login fails, you may be booted to the wrong system: follow the troubleshooting directions in the section called "Troubleshooting Boot Problems", later in this chapter.

4. Now boot the next client, and then proceed to "Adding a Local Disk" if you need to.

Troubleshooting Boot Problems

Are You Booted to the Wrong System?

If you did not interrupt the client's boot process as described earlier in this chapter, and you did not remove or turn off any disk drive that was attached to the client and contained a bootable system, then you are probably booted to the system on the disk.

Diagnose and correct this as follows:

1. Log in.
2. Enter the command:

   ```
   /bin/getcontext
   ```

   - If the system's response contains the word `remoteroot`, you are probably booted to the right system.

     (If this client is a member of more than one cluster, you could still be booted to the wrong server. You can check the names of the members of the current cluster by entering the command `cnodes`. For simplicity's sake we recommend that a client be a member of only one cluster.)

   - If the response is

     ```
     not found
     ```

     or contains the word `standalone` or `localroot`, then you are booted to the wrong system. Follow directions under "Tracking Down the Problem" on the next page.

Tracking Down the Problem

Take the following steps if the client is either booted to the wrong system, or not booted at all.

1. ■ If the client is booted to the wrong system, do the following:

   a. Make sure that no one else is logged into the cluster client system.

   b. Enter the command:

   ```
   /etc/shutdown -h 0
   ```

   c. When you receive the "halted" message, turn the cluster client's power off and on and interrupt the boot (as described under "Preparing To Add a Series 300/400 Client" earlier in this chapter).

   ■ If the client has not yet booted at all, try to boot it again now.

   Turn the client's power off and on and interrupt the boot (as described under "Preparing To Add a Series 300/400 Client" earlier in this chapter).

   If you do not see the cluster server's name on the line beginning `LAN ...` , turn power off and on once more to restart the boot ROM. See if the cluster server entry appears this time.

2. If you still do not see the cluster server entry, log in to the cluster server and check that `rbootd` is running on the server, for example:

   ```
   ps -ef | grep rbootd
   ```

   Also check the log file `/usr/adm/rbootd.log`.

3. If `rbootd` is running, you may have entered an incorrect station (link level) address into SAM. This address is now stored in the file `/etc/clusterconf`. You can check the format of entries in `/etc/clusterconf` with the `cccK(1M)` command (usually restricted to the superuser).

4. If `ccck` returns an error, you need to fix the item in error, and this may mean deleting the client and adding it back correctly. See Chapter 7, "Removing and Renaming Cluster Clients". If only the station address is wrong, however, you can simply change it in `/etc/clusterconf` (using the editor of your choice) and reboot the client.

   `ccck` refers to the station address as the "machine ID", and if the entry is incorrect in form (missing a digit, for example), you'll see:

   `Bad machine ID at line` *n*

   where *n* is the number of the line that contains the bad station address.

   (If the form of the station address is right, the address itself could still be the wrong one. `ccck` will not catch this.)

5. If something other than the station address is wrong (or you still don't know what is wrong), verify all the information you've entered about this client.

   Go back to Chapter 4, "Setting Up a Cluster", and check every item. If you find that you have entered an item incorrectly, use SAM to remove the client from the cluster and add it back correctly. See Chapter 7, "Removing and Renaming Cluster Clients".

6. If all the information you entered into SAM seems to be correct, the boot could be failing because of a LAN problem (broken or disconnected LAN, bad connection, unusually heavy LAN traffic) or a software incompatibility that is causing the client to panic.

   See Chapter 4, "Diskless Cluster Problems", in *Solving HP-UX Problems*, HP part number B2355-90030, for more details.

What To Do Next

You have now created (or expanded) and booted your cluster.

Tasks you may now need to do include:

- Adding disk drives to some of the clients.

  Local disk drives (drives attached to a client rather than to the server) can have any of the following uses:

  - Local swap.

    This means that the client swaps to its own local disk, rather than to the server's disk space.

  - Distributed swap.

    This is similar to local swap, except that one or more other clients, as well as the client that has the disk, swap to the local disk.

  - Shared file space.

    A disk attached to a client may contain a file system (but not the `root` (`/`) file system, nor system directories such as `/usr`, which must be on the server). This locally mounted file system is not private to the client to which the disk is attached, but is visible cluster-wide.

  You can use SAM to add a local disk, to configure local and shared swap, and to mount a local file system. See Chapter 12, "Adding Peripherals to a Cluster", for more information.

  "Adding a Local Disk", later in this chapter, contains a cookbook procedure for adding a disk drive to a cluster client.

- Adding other local peripherals, such as printers and tape drives.

  See Chapter 12, "Adding Peripherals to a Cluster".

- Adding users and groups.

  See Chapter 13, "Managing Users in a Cluster".

- Backing up the system.

  See Chapter 9, "Backing Up Files in a Cluster".

Adding a Local Disk

If you need to add a local disk to a new cluster client, and you're already familiar with the task, the following outline should serve to remind you of the major steps.

Note This is for quick reference only: if you have not connected a disk drive to a cluster client before, read the full explanation and example in Chapter 12, "Adding Peripherals to a Cluster".

Caution If you are connecting a disk drive that is to be used for local swap and that disk drive is connected to an E/ISA card, you *must* configure the card before the client attempts to swap to the disk. Otherwise the client will panic.

There's a more detailed note about this in Chapter 12, "Adding Peripherals to a Cluster", under "Example: Adding a Local Disk". For complete instructions on configuring E/ISA cards, consult *Installing Peripherals*, HP part number B1864-90011.

5

1. Connect the disk drive to the cluster client.

   Directions for each type of disk are in the *Installing Peripherals* manual.
2. Log in to the *cluster client* as superuser.

   (Remember that the superuser login is valid for every machine in the cluster, which means that the superuser password you have established on the server will work on every cluster node.)
3. Run SAM and get to the `Add a Hard Disk` screen.
4. Use this screen to configure swap and/or a file system.
5. If necessary, let SAM reconfigure the kernel reboot the client to which you have added the disk (SAM will prompt you if you need to do this).

6. If you have configured swap on the disk, and you want other clients to swap to this swap area, too (that is, if you want this client to be an **auxiliary swap server**), do the following:

   a. Log in to the *cluster root server*.

   b. Run SAM and select:

      `Cluster Configuration` .

Caution This is for quick reference only. You can get into trouble if you reboot clients in the wrong order while changing swap servers. If you have not already read the section on "Setting Up Swap to an Auxiliary Swap Server" in Chapter 12, "Adding Peripherals to a Cluster", do so now.

---

   c. Highlight the name of the *swap client (the client that is to swap to this swap server)* and choose

      `Modify Swap Server...`

      from the `Actions` menu.

   d. Highlight the name of the *swap server (the client to which you have just added the disk)* and press (OK).

      For example, if you have attached a swap disk to a client named `client1` and you want `client2` to swap to this disk, then change `client2`'s `Swap Server` to `client1`.

   e. Turn power off and on for each client whose swap server has changed, and let the client reboot.

      The change will not take effect until you do this.

6

Connecting the Cluster to Another Network

We recommend that you set up your cluster on a small, dedicated LAN (see Chapter 3, "Setting Up the Network"). This chapter explains how to connect the cluster LAN to the outside world (or simply to another network).

Note Before you do anything described in this chapter, you must first do the following:

1. Study the networking documentation (see Chapter 3, "Setting Up the Network").
2. Set up the cluster network (see Chapter 3, "Setting Up the Network").
3. Create the cluster (see Chapter 4, "Setting Up a Cluster")
4. Add cluster clients (see Chapter 5, "Adding Cluster Clients").

Recommendations

A cluster is itself a network. However, you may want the cluster to communicate with other machines which are not part of the cluster.

This means that one of the machines in the cluster will be a LAN **gateway**: that is, it will have two (or more) LAN cards, one connecting it to the cluster and the other(s) connecting it to the other network(s).

If you need to connect your cluster to another network, Hewlett-Packard recommends that you use the cluster server as the gateway, not one of the cluster clients.

Configure the networks as follows:

- Put all the cluster nodes (the cluster server and all the cluster clients) on one LAN.
- Configure the cluster server as the LAN gateway.
- Make the **official host name** in the `/etc/hosts` file the same for both LAN cards.
- Supply a unique **alias** in `/etc/hosts` for each card.

Example: Connecting the Cluster to Another Network

Your configuration should look like something like this:

```
 ----------        ----------       ----------
| 192.1.1.2|      | 192.1.1.3|     | 192.1.1.4|
|          |      |          |     |          |
| "client1"|      | "client2"|     | "client3"|
 ----------        ----------       ----------
      |                 |                |
      |                 |                |
-------Cluster LAN (192.1.1 Network)-----------
                        |
                 Cluster LAN card
                        |
                   ----------
                  | 192.1.1.1|
                  | "server" | Cluster Server/
                  |          | Gateway Node
                  | 192.2.2.1|
                   ----------
                        |
               Non-Cluster LAN card
                        |
-------Non-Cluster LAN (192.2.2 Network)-------
```

This picture shows two networks (network addresses 192.1.1. and 192.2.2).

The cluster network includes three cluster clients (`client1`, `client2` and `client3`) and the cluster server, `server`.

`Server` is a gateway node, connecting the cluster network (192.1.1) to the other (192.2.2) network.

How To Edit Networking Files for the Sample Network

To make the two networks co-operate efficiently, you need to do the following:

1. Create the cluster.

   You should have done this already. If you haven't, go back to Chapter 4, "Setting Up a Cluster".

   Do not do either of the following steps until after you have created the cluster and added clients.

2. Modify `/etc/hosts`.

   The networks represented on the previous page should be described in `/etc/hosts` as follows:

(Internet Address)	*(Official Host Name)*	*(Alias)*
192.1.1.1	server	dserver
192.1.1.2	client1	
192.1.1.3	client2	
192.1.1.4	client3	
192.2.2.1	server	dgateway

3. Modify `/etc/netlinkrc`.

Note Do not change `/etc/netlinkrc` until after you have created your cluster and added cluster clients.

To reflect the configuration shown under "Example: Connecting the Cluster to Another Network", you would modify `/etc/netlinkrc` as follows:

a. Find the statement

```
case $NODENAME in
    *) ifconfig lan0 'hostname' up
    ;;
esac
```

b. Insert text so that the statement in step 1 now reads:

```
case $NODENAME in
     $ROOTSERVER) ifconfig lan0 dserver up;
                  ifconfig lan1 dgateway up;
    *) ifconfig lan0 'hostname' up
    ;;
esac
```

c. Find the statement

```
case $NODENAME in
     *) # comment
        ;;
```

d. Insert text so that the statement in step c. now reads:

```
case $NODENAME in
     *) # comment
     $ROOTSERVER);;
     *) /etc/route add default dserver 1
        ;;
```

6

7

Removing and Renaming Clients

This chapter explains how to remove and rename cluster clients.

Caution You cannot change the name of the cluster server nor can you remove it from the cluster.

7

Removing Cluster Clients

This section explains how to remove cluster clients. (You cannot remove the cluster server).

Do this on the cluster root server.

1. Log in to the cluster server as superuser.
2. If you are planning to remove an **auxiliary swap server** (a client with a local disk that other clients are using for swap) shut down the swap clients now and turn them off.
3. Shut down all the clients you intend to remove.

   (See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", if you need help.)
4. Get to the `Cluster Configuration` screen in SAM.

   You'll see a list of cluster nodes: the cluster server and all the clients.
5. Highlight the name of each client you want to remove.
6. From the `Actions` menu, choose either `Remove Cluster Clients and their files` or `Remove Cluster Clients, keep files`.
   - If you choose to remove files, SAM will check the file system for all context-dependent files that contain an element with the cluster client's name, and will remove those elements.
   - If you choose to keep files, these client-specific elements will stay in the cluster server's file system.

   Choose `Remove Cluster Clients and their files` unless you have a specific reason to leave the client-specific elements on your system.

   (See Chapter 2, "Understanding Clusters", for an explanation of context-dependent files.)

7. Make sure you have shut down all the clients you are going to remove, and any clients that were swapping to their disk space; then choose (Yes) to confirm that you really want to remove these clients.

   If you have opted to remove the client-specific CDF elements, you'll see a message that a background script is running. You can leave this screen, and exit SAM altogether if you like.

Caution If, after choosing (Yes), you decide to remove more clients, or re-add the client you have removed (or add a different client with the same name), you must wait until the script has finished. The script will write a message at the end of `/tmp/cluster.log` when it's finished.

---

8. If you have removed a swap server, SAM will reconfigure its swap clients to swap to the root server.

   If you want these clients to swap to another client's disk space instead, you can change swap servers now: choose

   `Modify Swap Server`

   from the `Actions` menu.

   (See "Setting Up Swap to an Auxiliary Swap Server" in Chapter 12, "Adding Peripherals to a Cluster", if you need more help.)

9. Exit SAM. 7

Rebooting Swap Clients

A **swap client** is a client that is swapping to another cluster node's disk space.

If you shut down any swap clients because you were removing a swap server, you can reboot the swap clients now. They will swap to the root server's disk space by default.

Caution

- If you have reconfigured swap clients to swap to another client's disk space, make sure the new swap server is up and running before you boot any swap client.
- A simple reboot is not enough: you must turn power off and on again on each client whose swap server has changed.

What SAM Does When You Remove a Cluster Client

When you remove a client from the cluster, SAM does the following:

1. Removes the client's entry from `/usr/sam/config/cnode.config`.
2. Removes the client's entry from `/etc/clusterconf`.
3. Removes this client's entries from all context-dependent files (except those in NFS file systems, if any), if you chose to `Remove Cluster Clients and their files`.
4. Removes this client's entries from the following ARPA configuration files:
   - `.rhosts` (in the root user's home directory)
   - `/etc/hosts.equiv`
   - `/etc/X0.hosts` (if it exists)

   (SAM does not remove the client's entry from the `/etc/hosts` file.)
5. If the client was an **auxiliary swap server** (that is, if other clients were swapping to its disk space) SAM changes the swap clients' entries in `/etc/clusterconf` to make them swap to the cluster root server's disk space.

Renaming Cluster Clients

The easiest way to rename a cluster client is to remove it and then add it again with a different name. Follow these steps:

1. If you have customized any of the client's elements in the cluster's CDFs, you may want to save them by copying them to a different name.

   Let's say you are going to rename `client1` to `george`, and that you have customized `client1's /etc/inittab`.

   To save the customized copy of `/etc/inittab`, you might decide to copy it to `/tmp/inittab.george`:

   ```
   mv /etc/inittab+/client1 /tmp/inittab.george
   ```

   Since this command makes explicit reference to `client1's` element of `/etc/inittab`, you can issue it on the cluster server or on `client1` with the same effect.

2. Run SAM on the cluster root server to remove the cluster client and all its elements in the cluster's context-dependent files.

   Follow the directions under "Removing Cluster Clients" in the previous section. Chose the option to `Remove Cluster Clients and their files`.

3. Change the client's name in your "hosts" database.

   For example, change the *name* field in the cluster client's entry in `/etc/hosts` to the new name. This will associate the client's internet address with the new name.

4. Look at the end of the file `/tmp/cluster.log` to make sure that the SAM script that removes client-specific elements of context-dependent files has finished running. (See "Removing Cluster Clients".)

5. Use SAM to add the cluster client under the new name.

   Follow the directions in Chapter 5, "Adding Cluster Clients".

6. If you saved any files in step 1, copy those files back to the context-dependent files where they belong.

   For example, if you had saved the client's version of `/etc/inittab` as `/tmp/inittab.george`, and then renamed the client to `george`, you would issue the command:

   ```
   mv /tmp/inittab.george  /etc/inittab+/george
   ```

8

Introduction to Cluster Administration

An HP-UX cluster is designed to look and act as much as possible like a single, multi-user computer, and for the end-user there is little difference between HP-UX on a terminal or workstation and HP-UX on a cluster node.

But for you, the system administrator, there are significant differences. Chapter 2, "Understanding Clusters" explains the characteristics that make a cluster different from a **standalone** system (a computer that is not part of a cluster). The present chapter explains how these characteristics affect system administration tasks in practice.

Differences Between Standalone and Cluster Administration

In the day-to-day administration of a cluster, the following questions are likely to crop up frequently:

- Where (on which cluster node) to perform a given task.
- How to use HP-UX commands and subsystems which are changed because you are in a cluster, or which are unique to clusters.
- In particular, how to work with context-dependent files, which present different contents to different cluster nodes.

This chapter provides guidance in these cases. Other important topics are dealt with in separate chapters, including:

- Backing up the cluster's files, covered in Chapter 9, "Backing Up Files in a Cluster".
- Booting and shutting down the cluster, covered in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
- Reconfiguring a cluster node's kernel, covered in Chapter 11, "Reconfiguring the Kernel for a Cluster Node".
- Adding peripherals, covered in Chapter 12, "Adding Peripherals to a Cluster".
- Managing users, covered in Chapter 13, "Managing Users in a Cluster" .
- Updating HP-UX to a new release and installing applications, covered in Chapter 14, "Updating a Cluster".
- Evaluating and creating applications, covered in Chapter 15, "Applications in a Cluster".

Cluster configuration and networking tasks are covered in chapters 3 through 7.

When and Where To Perform System Administration Tasks

Table 8-1. Routine Tasks: When and Where to Perform Them

What	When	Where
Configure cluster	Set-up	Root Server (1)
Add clients	Cluster set-up, expansion	Root server (2)
Remove clients	Cluster restructuring	Root server (3)
Boot server	Set up, maintenance	Root server
Boot client	Set up, maintenance	Client
Set time and date	Initial boot	Any cluster node (4)
Create first full archival backup	After first boot	Root server (5)
Do incremental backup	Daily	Root server (5)
Do full backup	Weekly	Root server (5)
Create recovery System	After first boot	Root server
Create new user accounts	After first boot, expansion	Any cluster node (4)
Check disk usage	Weekly	Root server
Set run-levels	Set-up	Root server, clients (6)

8

Numbers in parentheses refer to notes on the next page.

Notes

1. See Chapter 4, "Setting Up a Cluster".
2. See Chapter 5, "Adding Cluster Clients".
3. See Chapter 7, "Removing and Renaming Cluster Clients".
4. You can do this on any cluster node (the server or any client) and the result will be global to the cluster.
5. To back up a cluster's files, you need to use special options to locate the contents of context-dependent files (CDFs). For example, use the `-H` option of the `fbackup(1M)` utility. For performance reasons, you should do backups on the cluster server.

   See Chapter 9, "Backing Up Files in a Cluster".
6. You can set run-levels individually for each member of the cluster, because `/etc/inittab` is a context-dependent file.

   For more information on setting run-levels see Chapter 4, "Controlling Access to the System", in the *System Administration Tasks* manual.

Table 8-2. Tasks Required by Specific Events

Event	What To Do	Where
Power going out soon	Shut down cluster. Turn power off for server and clients	Root server, clients, peripherals (1)
Bringing cluster back up	Boot server, then clients	Root server, clients (2)
Local powerfail on root server	Turn off then reboot clients	Clients (3)
Local powerfail on auxiliary swap server	Turn off then reboot clients	Clients (3)
Local powerfail on client that is not auxiliary swap server	Turn off then reboot client	(3)
Root server maintenance needed	Shut down the cluster, power down root server	Root server (1)
Auxiliary file server maintenance needed	Get users out of file system, power down auxiliary file server	Auxiliary file server (8)
Auxiliary swap server maintenance needed	Shut down clients swapping to auxiliary server, power down auxiliary swap server	Auxiliary swap server (8)
Maintenance needed on client that is not an auxiliary server	Halt and power down client	Client (4)
Need to send message to all cluster users	Use `cwall(1m)`	Any cluster node (5)
Need to send message to all users of one cluster node	Use `wall(1m)`	Affected cluster node
Files accidentally deleted	Recover files from backup	Root server

Table continues on the next page; numbers in parentheses refer to notes following the table.

8

Tasks Required by Specific Events (continued)

Event	What To Do	Where
File system corrupted	Use `fsck (1M)`, recovery system, archives as appropriate	Root or auxiliary server (6)
Operating system corrupted (or other special circumstances)	Re-install operating system, re-create cluster	Root server (9)
New user arrives	Use SAM to create account	Any cluster node (5,10)
User leaves	Use SAM to delete account	Any cluster node (5,10)
Need to create/change group	Use SAM	Any cluster node (5)
System clock wrong	Set clock ahead/back	Any cluster node (5)
Root server panics	Diagnose problem using the manual *Solving HP-UX Problems*, HP part number B2355-90030. Reboot server. Recover files. Reboot clients.	Root server, clients
Adding a peripheral	Follow directions in Chapter 12, "Adding Peripherals to a Cluster". Use SAM if possible.	Cluster node peripheral is connected to
Changing kernel configuration	Rebuild/reinstall kernel	Cluster node that will use kernel (7)

Numbers in parentheses refer to notes on the following pages.

Notes

1. See "Shutting Down a Cluster" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
2. See "Booting a Cluster" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
3. **Powerfail** means a power failure that halts the computer. **Local powerfail** means that the power goes out for a particular node, but not for the whole cluster.

Caution TURN OFF all computer equipment affected by a power failure until power is completely restored. An electrical surge as power is coming back on could seriously damage hardware that has been left turned on.

- When a local powerfail occurs on a cluster root server, all the clients will panic. Make sure you switch off the root server and all other equipment that no longer has power.
- When a local powerfail occurs on an **auxiliary swap server**, the clients swapping to the auxiliary swap server will panic. Make sure you switch off the auxiliary server and all other equipment that no longer has power.

  An auxiliary swap server is a client to whose disks other clients are swapping.
- When a local powerfail occurs on an **auxiliary file server**, the locally mounted file systems will be unavailable until the auxiliary file server comes back up. Other cluster nodes that are not affected by the power failure will continue to function. Make sure you switch off the auxiliary server and all other equipment that no longer has power.

  An auxiliary file server is a client whose local disk is used for a file system but not for shared swap.
- When a powerfail occurs on a client that is not an auxiliary swap or file server, other cluster nodes that are not affected by the power failure will continue to function. Make sure you switch off the client and all other equipment that no longer has power.

To bring the cluster clients back up after a powerfail on the root server or an auxiliary swap server, turn the server back on and wait for it to reboot, then reboot the clients.

4. See "Shutting Down a Cluster Client", in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
5. You can do this on any cluster node (the root server or any client) and the result will be global to the cluster.
6. - To run `fsck(1M)` on the root file system, you must be logged in to the root server.
   - To run `fsck` on a locally mounted file system, you must be logged in to the auxiliary file server (the client to which the disk is attached).

     See Chapter 12, "Adding Peripherals to a Cluster" for more information on locally mounted file systems.
7. See Chapter 11, "Reconfiguring the Kernel for a Cluster Node".
8. See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
9. If you have to re-install the operating system, you must rebuild the cluster from scratch:
   a. Do a full back-up of the entire system.
   b. Re-install the operating system on the cluster server, following directions in the manual *Installing and Updating HP-UX*.
   c. Re-configure the cluster server, using Chapter 4, "Setting Up a Cluster", in this manual.
   d. Add back the cluster clients, following directions in Chapter 5, "Adding Cluster Clients", in this manual.
   e. Recover the files you backed up.
10. See Chapter 13, "Managing Users in a Cluster".

Table 8-3. Miscellaneous Tasks: Where to Perform Them

What	Where
Mount/unmount file systems on HFS	Cluster node that the disk is attached to (1)
NFS mount/unmount	Server, clients (2)
Create file system	Cluster node that the disk is attached to (1)
Update HP-UX	Server (3)
Install or update applications	Server (3)
Remove filesets	Server
Configure LP spooler	Server (4)
Set up, use UUCP	Server (5)
Modify system files	(6)

Notes

1. You must use file-maintenance commands such as `mount(1M)` while logged in to the node to which the disk is attached. See "Commands Whose Use is Restricted in a Cluster", later in this chapter.
2. You can do this from any cluster node, and the result will be global to the cluster.

   If the NFS mount has been made from a client and the client is shut down, the mount will continue to function normally for the remaining cluster nodes, and will function for the client once it is rebooted: there is no need to remount the file system.
3. See Chapter 14, "Updating a Cluster" for directions for updating HP-UX and installing and updating applications.

4. ■ The spooler runs only on the cluster root server.

   ■ A printer can be attached to any cluster node and can be either spooled (via the root server) or unspooled.

     □ Unspooled printers are available only to users of the local node.
     □ Spooled printers are available to all users of the cluster.

   See Chapter 12, "Adding Peripherals to a Cluster" for more information.

5. UUCP is supported only on the cluster root server.

   ■ All UUCP connections must be on the cluster server.

   ■ UUCP transfers can be initiated from any cluster node.

6. When you configure an HP-UX system as a cluster server, many system files are converted to context-dependent files, whose contents differ depending on which member of the cluster is looking at them. Thus when you modify a system file, you may be modifying it for the whole cluster, for some members of the cluster, or for all members (**nodes**) of the cluster, depending on what type of file it is.

   ■ If you modify a **node-specific context-dependent file** such as `/etc/inittab`, you will, by default, modify only those contents of the file that pertain to the cluster node you're logged in to when you edit the file.

     A node-specific context-dependent file has different contents (**elements**) for individual nodes in the cluster. `/etc/inittab` has an element for each node because the information may need to be different from one node to the next.

   ■ If you modify a system file that is not a context-dependent file, you can edit it from any cluster node, and the result will be global to the cluster.

   The section titled "Finding System Context-Dependent Files (CDFs)", later in this chapter, explains how to determine whether a given system file is context-dependent or not. Chapter 2, "Understanding Clusters" explains how context-dependent files work, and later sections of the present chapter provide examples and guidelines.

Using HP-UX Commands in a Cluster

This section lists commands and options of commands that are useful in administering an HP-UX cluster, as well as commands that work differently in a cluster. Some commands are restricted to certain nodes; these are listed at the end of the section.

Many of the commands act on, or are affected by, **context-dependent files**.

A Key Concept: Context-Dependent Files

In order for different machines to use the same file system, a given file may need to have different contents depending on which particular member of the cluster is looking at it. This type of file is known as a **context dependent file** or **CDF**.

A CDF is really a special kind of directory, containing files which are called **elements**. Each element name should match some **context attribute** of one or more cluster nodes.

When the HP-UX file system finds a match between a *context attribute* of a given node (server or client) on one hand, and an *element* of a context-dependent file on the other, it makes the file visible by default on the node in question; on other nodes it reports `not found`, although you can override this protection by using special options to commands, as the following pages show. (See "Using the ll Command with the -H Option" and "Working with Context-Dependent Files: Examples", later in this chapter.)

As you configure a cluster, HP-UX utilities automatically build the CDFs that the system will need, but you may also need to create your own CDFs as you add new applications to the cluster, if those applications are intended to run on certain nodes and not on others.

The *HP-UX Reference*, HP part number B2355-90033, and Chapter 2, "Understanding Clusters", in this manual, provide details of how context-dependent files work, and there are examples later in the present chapter.

Commands Specific to Clusters

The following commands perform operations that are needed only in a cluster.

(The numbers in parentheses immediately following the command names refer to the relevant sections of the *HP-UX Reference*.)

`showcdf(1)`

Shows which element of a context-dependent file will be used on the current cluster node.

For example, on system `server`, the command

```
showcdf /etc/checklist
```

would produce this output:

```
/etc/checklist+/server
```

The `+` shows that `/etc/checklist` is really a hidden directory (formally referred to as a **context-dependent file**) containing the **node-specific** file `server`, whose contents are `server's` version of `/etc/checklist`. (**Node-specific** means that the cluster node `server`, and only `server`, will see this version of `/etc/checklist` by default.)

We normally use the term **element** to describe a file inside a CDF: thus `server` is an **element** of the context-dependent file `/etc/checklist`.

If you use the wild card (*) to examine a group of files, use the `-c` option of `showcdf` to avoid cluttering up the display with the names of files that are not context-dependent files.

To see the names of the current node's elements of all the context-dependent files in `/etc`, enter:

```
showcdf -c /etc/*
```

`cnodes(1)`

Lists the cluster nodes in a cluster.

To list all cluster nodes in the cluster:

```
cnodes
server client1 client2
```

To get the cluster node name of the local system:

```
cnodes -m
client1
```

To get the cluster node name of the cluster's root server:

```
cnodes -r
server
```

To list the status of all cluster nodes configured in the `/etc/clusterconf` file:

```
cnodes -alC
CNODE       ID SWAP SITE
========== === ==========
server       1 server     ROOTSERVER
client1      2 client1
client2*     3 server
client3      4 client1
```

In the above example, the "*" by `client2` indicates that it is not currently booted to the cluster. This could indicate that it is not booted at all, or it could be booted from its own disks (in standalone mode). The computer `client1` is swapping to its own disks (this is called **local swap**) and is also serving as a **swap server** to `client3` (that is, `client3` is swapping to `client1`'s disk space, not to `server`'s).

8

`getcontext(1)`	Shows the context of the current cluster node. The **context** is an ASCII string that identifies a particular cluster node. You need to know the context when creating context-dependent files (see Chapter 2, "Understanding Clusters" for details).
`makecdf(1M)`	Creates a new CDF, or converts an existing file or directory to a CDF. See "Creating a CDF" later in this chapter.
`cfuser(1M)`	Cluster-wide version of `fuser(1M)`.
`cps(1)`	Cluster version of `ps(1)`. Optionally reports processes on a node-by-node basis. See `ps(1)` in the *HP-UX Reference*.
`cwall(1M)`	Cluster version of `wall(1M)`. Use `cwall` when you want to broadcast a message to all users of all cluster nodes. You can use `cwall` from any cluster node, and everyone using the cluster will get the message (you must be superuser to override users' protections). Use `wall` on a given cluster node when you want the message to go only to users of that particular node. As with other utilities that write directly to the screen, use `wall` and `cwall` only when the message is urgent enough to warrant interfering with X-Windows and VUE displays.
`csp(1M)`	Starts the cluster server processes needed by each cluster node to communicate within the cluster.

Refer to the *HP-UX Reference*, HP part number B2355-90033, for more information on all of these commands.

Cluster Options for Common Commands

Some HP-UX commands have special options for use in a cluster. These cluster options are listed below.

`-c` This option indicates to certain commands that those commands should operate on the whole cluster, not just the current machine.

For example,

```
who -c
```

will show you all the current users of all the nodes in the cluster.

The following commands support this option:

- `last(1)` (and `lastb`)
- `users(1)`
- `who(1)`

`-H` *or* `H` This option means, roughly, "expose **H**idden directories (context-dependent files) to the operation of the command"—but check the *HP-UX Reference* for the exact meaning in each case.

This option is available with the following commands:

- `chmod(2)`
- `ls(1)` (and `ll`, etc.)
- `tar(1)`
- `test(1)`
- `pwd(1)`
- `fbackup(1M)`
- `find(1)`

8

`-hidden` This option is available with the `find(1)` command. It finds all elements of context-dependent files.

`-l` Indicates to certain commands that those commands should operate on the node on which it's executed (`l` is for "local").

Available with the following commands:

- `sync(1M)`
- `mount(1M)`
- `df(1M)`
- `bdf(1M)`

- `sync` causes all file system updates to be written to disk.

  The `-l` option causes `sync` to flush all the buffers on the node on which it's executed and writes their contents to disk.

- `mount` without options prints the system table of mounted file systems and disks, `/etc/mnttab`.

  The `-l` option causes `mount` to print information only about file systems on the current node's local disks.

  (Pathnames for devices and file system mount points are fully expanded in `/etc/mnttab`, showing the "hidden" elements of context-dependent files.)

- `df` and `bdf` report the amount of free disk space in a given file system.

  The `-l` option causes `df` and `bdf` to print information only about file systems on the current node's local disks.

`-L` Available with the following commands:

- `mount(1M)`
- `df(1M)`
- `bdf(1M)`

Similar to `-l`, but indicates the command should provide information about NFS mounted file systems as well as locally mounted file systems.

Example: Using the ll Command with the -H option

Suppose you have a file `/users/patrick/myfile` which is actually a context-dependent file (CDF) with a single element specific to the cluster node `client1`.

This means that `ls`, `ll`, etc, will not show you the file on clients other than `client1`, nor on the server, unless you use the special option `-H`.

The `ll` command will produce varying output, depending on whether or not you use the `-H` option and which node you are logged in to when you execute the command:

on client1

```
ll /users/patrick/myfile
-rw-rw-rw-   1 patrickd users          0 Mar 26 17:06 /users/patrick/myfile
```

on client2

```
ll /users/patrick/myfile
/users/patrick/myfile not found

ll -H /users/patrick/myfile
total 0
-rw-rw-rw-   1 patrickd users          0 Mar 26 17:06 client1
```

As you can see from the output of `ll -H`, `/users/patrick/myfile` is logically a directory, containing the file `client1`. But usually we would speak of `/users/patrick/myfile` as a **context-dependent file**, which has the element `client1`. The full pathname would be written as `/users/myfile+/client1`, and the file could be used under that name on any node in the cluster by any user with the appropriate permissions.

Directories can also be context-dependent: that is, the contents of an entire directory can vary depending on which node in the cluster is looking at it.

Try entering

```
ls -H /usr
```

on any cluster node. You'll see something like this:

```
adm+       doc/       lib/      netdemo/   preserve/   sysver/
bin/       etc/       local/    netls+     pub/        tmp/
boot/      include/   lotus/    nettest/   sam/        tsm/
contrib/   keysh/     mail/     news/      softbench/
diag/      kmail/     man/      old/       spool/
```

A name followed by a plus sign indicates a context-dependent file or directory; in this case, the plus signs indicate context-dependent directories. You might want to compare this to the output of the command:

```
lsf /usr
```

If you then enter the command:

```
ls -H /usr/adm
```

you will see a list of the elements of the context-dependent directory `/usr/adm`. Each element should correspond to the name of a node in the cluster. This type of directory is called **node-specific**.

Commands and Scripts that Work Differently in a Cluster

Some HP-UX commands and scripts work differently in a cluster. These differences are described below.

`/etc/rc`
: The `/etc/rc` script performs slightly different functions depending on whether it runs on the cluster server, a cluster client, or a standalone machine. For example, the remote boot daemon (rbootd) is started only on the cluster server and on clients that will serve as auxiliary swap or file servers to other clients in the cluster.

`/etc/reboot`
: Performs a cluster-wide reboot when executed on the cluster root server. Reboots all clients swapping to an auxiliary swap server (a client with disk space used for swapping by other clients), when the swap server is rebooted.

  When you execute `/etc/reboot` on a cluster client that is not an auxiliary swap server, only that client is rebooted.

  See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

`/etc/shutdown`
: Performs an orderly shutdown of the entire cluster when executed from the cluster root server. Reboots all clients swapping to an auxiliary swap server when the swap server is rebooted (using `shutdown` with the `-r` option).

  When you execute `/etc/shutdown` from a cluster client that is not a swap server, only that client is shut down.

  See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

In addition, line-printer spooler commands work a little differently in a cluster, so as to allow printers attached to a cluster client to be spooled. See the section called "Example: Adding a Spooled Printer to a Cluster Client" in Chapter 12, "Adding Peripherals to a Cluster".

Commands Whose Use Is Restricted in a Cluster

Commands that operate on a file system will work only on the cluster root server, or on a client which has a local disk drive on which there is a file system (or on which you are building a file system). Such a client is called an **auxiliary file server**.

The following file system commands must be executed on the cluster node (root server or auxiliary file server) that has the disk on which the file system in question resides.

- `fsck(1M)`
- `fsclean(1M)`
- `fsdb(1M)`
- `fuser(1M)`
- `mediainit(1)` (for a disk)
- `mkfs(1M)`
- `mount (1M)`
- `newfs(1M)`
- `tunefs(1M)`
- `umount(2)`
- `mkrs(1M)`

The following command will work *only on the cluster root server*.

- `update(1M)`

Cluster-Specific Files

/etc/clusterconf

`/etc/clusterconf` is the cluster configuration file. It is used by utilities and system processes to obtain information about the cluster nodes.

Caution `/etc/clusterconf` must *always* exist in a cluster, and *never* on a standalone system.

The SAM utility creates `/etc/clusterconf` when you configure the cluster server, and modifies it as you add and remove clients, change client swap locations, etc.

If you suspect that there is something wrong with `/etc/clusterconf`, you can use the `ccck(1M)` command to check for syntax errors.

The following table shows the layout of `/etc/clusterconf`. The structure is the same for each line after the first.

Note Do not remove the first non-comment line (comment lines begin with a pound sign "#"): the system may not cluster without this first, special line.

Each of the remaining lines in `/etc/clusterconf` has six fields. Each field is separated by a colon (:).

Field Number	Description
1	Station address (link level address) of the LAN card. On a Series 300/400 you can get the link level address from the boot ROM, or from the **landiag** utility. (See Chapter 4, "Setting Up a Cluster").
2	Cluster node number. This is a unique but arbitrary number between 1 and 255. SAM will always assign 1 to the cluster root server, and will assign numbers sequentially for the other cluster nodes. *DO NOT change these numbers once they have been created by SAM!* This could cause problems in communication between the computers in your cluster.
3	Cluster node name. This is a unique name of no more than 8 characters. It should be the same as the name set by **uname -S** and should be the same as the first portion of the ARPA host name. You cannot use the names **default**, **HP-PA**, **UNKNOWN** or anything beginning with **HP-MC**.
4	Type of cluster node. This can be: `r = cluster root server` `c = cluster client` *(includes auxiliary servers)*
5	Cluster node number of the current node's swap server.
6	Number of Cluster Server Processes (CSPs) to run. This will probably be a number between 4 and 8 on the cluster root server and auxiliary servers. SAM assigns 1 to all new clients, and raises the number to 4 when you add a disk for shared swap or a file system.

An /etc/clusterconf file will look something like this:

```
0800090039dd: /* clustercast address.  Do not remove. */
0800090039dd:1:server:r:1:8
080009000565:2:client1:c:1:1
08000900297c:3:client2:c:1:1
```

CDFinfo Files

Your system contains a set of files, /system/*/CDFinfo, that you must never remove or modify in any way. These files contain the specifications for all the context-dependent files that SAM builds when you create a cluster or add clients. Update(1M) also uses the CDFinfo files when you add or update software.

Finding System Context-Dependent Files (CDFs)

Before you modify a system file (such as /etc/inittab) you need to know whether it is a context-dependent file, otherwise you may make changes you didn't intend, or fail to make changes you thought you were making.

This is because a context-dependent file can have different contents (**elements**) for different members of the cluster. /etc/inittab, for example, has a separate element for each member of the cluster.

Directories can also be context-dependent. The directory /usr/adm is an example.

Thus a system file can be a regular file, or a context-dependent file, or it can be a regular file in a context-dependent directory; it could even be a context-dependent file in a context-dependent directory. (All these possibilities also exist for files that are not system files: CDFs are really directories, and they can be nested inside each other just as regular HP-UX directories can.)

`/tmp/cluster.log` shows which system files and directories SAM converted to context-dependent files when you configured the cluster. If you did not print a copy of it after you configured your cluster server, print one now.

If you don't have a copy of `/tmp/cluster.log`, you can check whether a particular file is context-dependent by means of the `showcdf(1)` command, for example:

```
showcdf /etc/inittab
```

On system `client1`, the response would be

```
/etc/inittab+/client1
```

`/etc/inittab` is a context-dependent file that has a separate element for each node in the cluster.

If the system responds with just the file name, this is not a context-dependent file. For example, the command,

```
showcdf /etc/rc
```

produces the output

```
/etc/rc
```

showing that `/etc/rc` is not a context-dependent file.

There's more about context-dependent files in Chapter 2, "Understanding Clusters", and later sections of the present chapter provide examples.

Finding All CDFs

To find all the CDFs in the system, enter:

```
find / -hidden -fsonly hfs -type H -print
```

Working with Context-Dependent Files: Examples

This section provides examples of some of the things you will commonly need to do with context-dependent files (CDFs). You may want to sit down at the computer and try some these operations yourself.

Remember that a CDF is really a special kind of directory and that the files this type of directory contains are called **elements**, one or more of which may match a **context attribute** of a given cluster node.

For a full explanation of what CDFs are, see Chapter 2, "Understanding Clusters", under "Context-Dependent Files"; if you've already read chapter 2 but need a brief refresher, you'll find one at the beginning of the section "Using HP-UX Commands in a Cluster", earlier in the present chapter.

Listing CDFs

Suppose you have the `/etc/inittab` file shown under "Context-Dependent Files" in Chapter 2, "Understanding Clusters", and are currently logged in to the cluster node named `server`. Using different options of the `ll` command, you would see the following responses (note the `-H` option, for "hidden"):

```
$ ll /etc/inittab
-rwxr-xr-x  1 root    other     725 Dec 26 07:45 /etc/inittab

$ ll /etc/inittab+
-rwxr-xr-x  1 root    other     725 Dec 12 07:45 server
-rwxr-xr-x  1 root    other     650 May 26 15:53 client1
-rwxr-xr-x  1 root    other     650 May 18 11:47 client2

$ ll -d /etc/inittab+
drwsr-xr-x  2 root    other     1024 Dec 26 07:30 /etc/inittab+

$ ll -H /etc/inittab   # this is the same as ll /etc/inittab+
-rwxr-xr-x  1 root    other     725 Dec 12 07:45 server
-rwxr-xr-x  1 root    other     650 May 25 15:53 client1
-rwx--xr-x  1 root    other     625 May 18 11:47 client2
```

8

The `showcdf` command displays the CDF element that is visible to the node on which you are logged in. If you are on system `server`, the command,

```
showcdf /etc/inittab
```

will produce this result:

```
/etc/inittab+/server
```

Looking at Files Inside a CDF

If you are on the cluster server (`server`) in the sample cluster, the command

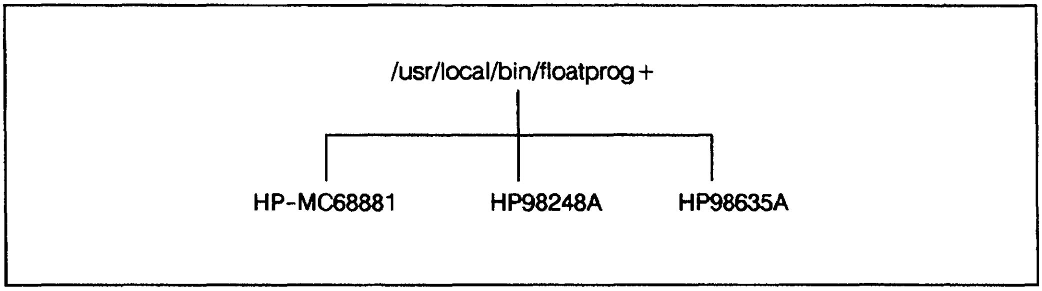
```
more /etc/inittab
```

will display the contents of the file `/etc/inittab+/server`. To look at the `inittab` file for system `client2` from the cluster server, enter:

```
more /etc/inittab+/client2
```

Creating a CDF

Suppose the Series 300 clients in the sample cluster shown in Chapter 2, "Understanding Clusters", frequently run a floating-point-intensive program: `/usr/local/bin/floatprog`. The systems have different floating point hardware and require a different version of the program to take advantage of this. A CDF to allow each Series 300 client in the sample cluster to execute the version of the floating point program best suited to its hardware configuration would look like Figure 8-1.



LG200181_005

Figure 8-1. CDF Structure Using Different Floating Point Hardware

Assume you created the original `floatprog` on `client1`, the machine with the HP MC68881 processor.

You have since created two additional versions of the program. These versions are in the files `/users/progs/fp48A` and `/users/progs/fp35`.

You can create a CDF and move the compiled versions of the program into it with the following commands:

```
makecdf -c HP-MC68881 /usr/local/bin/floatprog
mv /users/progs/fp48a /usr/local/bin/floatprog+/HP98248A
mv /users/progs/fp35 /usr/local/bin/floatprog+/HP98635A
```

(See `makecdf(1)` in the *HP-UX Reference* for more information on `makecdf` and its options.)

If you tried to run `floatprog` from a cluster node without any of this floating point hardware you would receive the message:

```
floatprog: file not found
```

The `default` context attribute might help here: if you had a version of the program that required no floating point hardware, you could place it in the CDF under the element name `default`, and it would be seen by any node that did not have `HP-MC68881`, `HP-98248A` or `HP-98635A` in its context. (But make sure you read and understand the cautions about mixing CDF element types later in this chapter.)

If you have a large group of files specific to a given cluster node or group of nodes, you may want to create a context-dependent directory.

Creating a CDF from a Regular File or Directory

You need to be aware of what happens when you create a CDF out of an existing regular file, and what happens when you make a CDF out of an existing *directory*. The results of the second operation are not analogous to the first.

File. If you make a cdf out of an existing **flat file** (regular HP-UX file), the contents of the flat file are copied to every element of the new CDF. We'll start with a flat file named `/users/patrick/tryit`, which contains the line:

```
You can try this yourself.
```

(on "client1")

```
ls -H /users/patrick/tryit
/users/patrick/tryit                     (file is not a CDF)
more /users/patrick/tryit
You can try this yourself.
makecdf /users/patrick/tryit
ls -H /users/patrick/tryit
client1   client3   client5              (file is now a CDF
client2   client4   server               with 6 elements)
more /users/patrick/tryit
You can try this yourself.
more /users/patrick/tryit+/client5
You can try this yourself.
```

(on "server")

```
more /users/patrick/tryit                (contents were copied
You can try this yourself.               to each CDF element)
```

Using the `makecdf` command (and defaulting all of the options), we have created a CDF named `/users/patrick/tryit+` with a separate element for each cluster node. The elements, by default, are named after the first item in each cluster node's **context string**: the **node name**. (See Chapter 2, "Understanding Clusters", in this manual, under "What Is Context", and the `makecdf(1M)` entry in the *HP-UX Reference*, for a full explanation.)

Directory. If you make a directory into a CDF, however, the files in that directory are *not* copied to all the elements of the new context-dependent directory. Instead, they are moved into a directory whose name is the first context element you specify on the command line (by means of the `-c` option), and empty directories are created for the other elements you specify.

If you use the `makecdf` command without specifying any elements (that is, without using the `-c` option), the files are moved into a directory whose name is the node name in the first entry in `/etc/clusterconf` (usually the cluster server).

For more information, see the `makecdf(1M)` entry in the *HP-UX Reference*.

Creating a New Element of a CDF from a Regular File

You can create a new element of a context-dependent file (CDF) by copying or moving a file to it. This is like creating a new file in an existing directory.

Suppose you have a CDF `/users/patrick/cdf.file` with the single element `client1` (`/users/patrick/cdf.file+/client1`).

You have a flat file (regular file) `/users/patrick/z` whose contents you would like to store in `cdf.file`, such that they can normally be seen only by the cluster node `server`. In other words, you want to create the CDF element `/users/patrick/cdf.file+/server` and place the contents of `/users/patrick/z` in it. You can do this as follows:

```
cd /users/patrick
mv z cdf.file+/server
```

This will work from any node in the cluster.

If you are logged in to the cluster node `server`, then you can do this more simply:

```
cd /users/patrick
mv z cdf.file
```

This will create the element `user/patrick/cdf.file+/server`, even though you have made no explicit reference to it. (This is called **autocreation**, which is explained in Chapter 2, "Understanding Clusters".)

What will *not* work correctly is:

`mv z cdf.file+` *(don't do this!)*

No matter what node you were on, this would add an element `z` to `cdf.file` (`/users/patrick/cdf.file+/z`), which would not be accessible by default from any node (unless you happened to have a node named `z`).

Moving a CDF

When you move or copy a CDF, you can lose the CDF structure if you're not careful. Suppose you have a context-dependent file /users/patrick/cdf.file with elements server and client1, and you are logged in to server. Now you do the following:

(On "server")

```
cd /users/patrick
showcdf cdf.file
cdf.file+/server
mv cdf.file zz
```

You have moved those contents of cdf.file that are specific to the node server (that is, the file cdf.file+/server), to the file zz, but zz is *not* a context-dependent file:

(On "server")

```
showcdf cdf.file
cdf.file: Inaccessible
ls -H zz
zz
ls -H cdf.file
client1
```

This may have been what you intended. If not, you can put things back the way they were:

(On "server")

```
mv zz cdf.file
ls -H cdf.file
client1 server
```

8

If what you really want to do is to move the whole of `cdf.file` (all the node-specific elements and the CDF structure itself) then you need to append a `+` to the source file name:

(on any cluster node)

```
cd /users/patrick
ls -H  cdf.file
client1 server
mv cdf.file+ zzy
ls -H zzy
client1 server
```

Removing a CDF

Caution Do not remove system CDFs (those that are created by SAM).

Context-dependent files (CDFs) are really directories, so in order to remove one you need to remove the contents and structure in the same way as you would remove a directory and its files.

Just as

```
rm -rf notwanted
```

gets rid of the directory `notwanted` and all its files, so

```
rm -rf notwanted+
```

gets rid of the CDF `notwanted` and all its elements.

Note the `+` escape character, which tells HP-UX you are addressing the full directory structure of the CDF.

If you omit the trailing `+`, only one element, corresponding to the node name or some other attribute of the cluster node you are logged in to, will be removed. For example, if you are logged in to system `server`, the command `rm -rf notwanted` will remove the element `server`. If no `server` element exists, another element matching `server's` context, if any, will be removed: `localroot`, for example. ("What Is Context" in Chapter 2, "Understanding Clusters", lists all the attributes in a cluster node's context in the order in which HP-UX commands look at them.)

Removing Elements from a CDF

You can remove elements from a context-dependent file (CDF), just as you can remove files from a directory.

Caution Do not remove elements of system context-dependent files (those that are created by SAM). The instructions that follow are only for CDFs you create yourself.

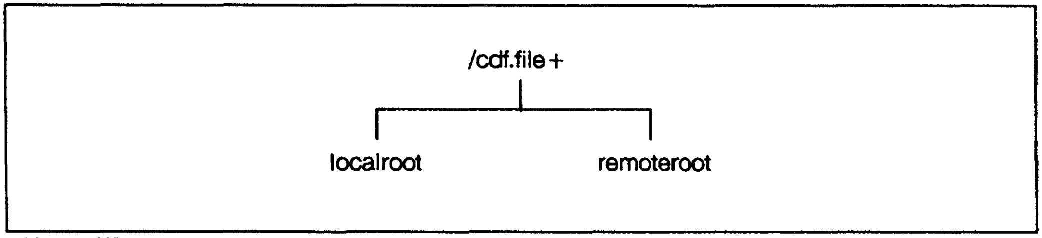
The `rm` command will remove the element of a context-dependent file that matches the context of the node you are logged in to.

For example, if you are logged in to a cluster node whose name is `server` and `/cdf.file` is a CDF containing the element `server`, then

```
rm /cdf.file
```

will remove the element `/cdf.file+/server`. We call the element `server` **node-specific** because it matches the context of only one node in the cluster (no two cluster nodes can have the same name).

But CDFs can have element names that match the context of more than one cluster node. For example, the CDF shown in Figure 8-2 has the elements `localroot` and `remoteroot`.



LG200181_007

Figure 8-2. CDF with "localroot" and "remoteroot" Elements

The `remoteroot` element of this file can be seen by all the clients in the cluster, since they all share the `remoteroot` context attribute (see "What Is Context" in Chapter 2, "Understanding Clusters").

If you remove the `remoteroot` element and then try to replace it, you may get an unexpected result.

If you are logged in to a cluster client whose node name is `client2`, the command,

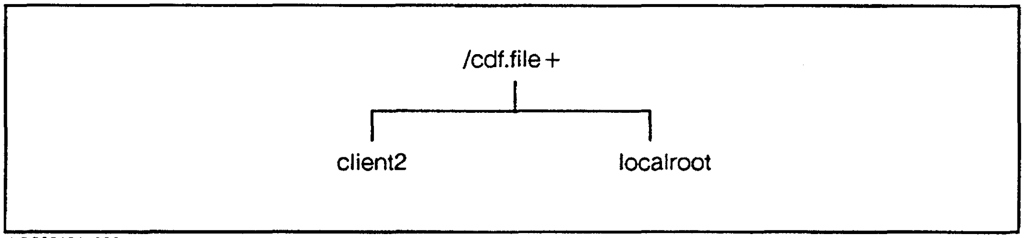
```
rm /cdf.file
```

will remove `/cdf.file+/remoteroot`. No client will now be able to see `/cdf.file` (unless the user specifies `/cdf.file+/localroot`, the full path name of the remaining element).

If, while still logged in to `client2`, you now do the following,

```
cp /tmp/newfile /cdf.file
```

`/cdf.file+/client2` will be created by the autocreation mechanism explained in Chapter 2, "Understanding Clusters". The result will be the CDF shown in Figure 8-3.



LG200181_008

Figure 8-3. CDF with Mixed Elements

If a user on `client1` now tries to list the file with `ls /cdf.file`, she will not find it.

This comes about because the autocreation mechanism creates elements named after the cluster node you're logged in to, unless you specify otherwise: the command, 8

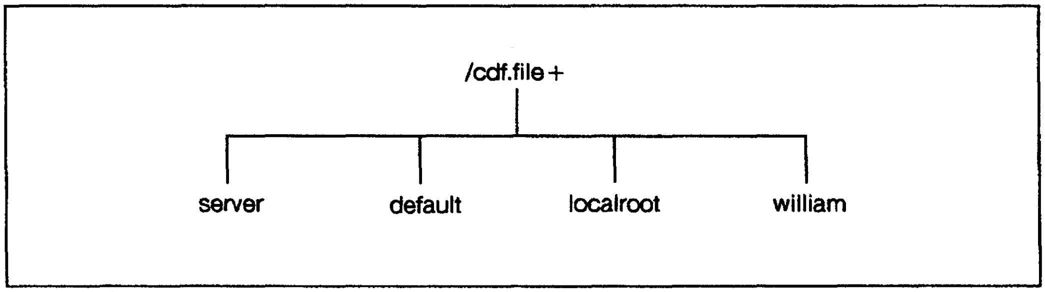
```
cp /tmp/newfile /cdf.file+/remoteroot
```

executed on `client2` (or any other cluster node), would have recreated the CDF shown in Figure 8-2.

Removing an Element from a CDF that Has Mixed Elements

Mixing types of elements in the same CDF can lead to confusion and unintended results, particularly when you remove elements.

Suppose for example you have the CDF shown in Figure 8-4.



LG200181_010

Figure 8-4. CDF with Mixed Element Types

The command,

```
rm /cdf.file
```

executed on the root server (whose node name happens to be `server`), would first remove the cluster-node-name element, `/cdf.file+/server`; next time it would remove the cluster-node-type element, `/cdf.file+/localroot`; and the third time it would remove the `default` element, `/cdf.file+/default`. ("What Is Context" in Chapter 2, "Understanding Clusters", lists all the attributes in a cluster node's context in the order in which HP-UX commands look at them.)

You would need to remove all three elements (`server`, `default` and `localroot`) if you wanted `/cdf.file` not to been by `server`. But if you wanted to remove only a specific element (`localroot`, for example, which is probably redundant), then you should specify the element explicitly:

```
rm /cdf.file+/localroot
```

This form of the command would remove the `localroot` element, and only that element, no matter which computer in the cluster you were logged in to when you issued the command.

Tips and Cautions

This section summarizes some of the most important points to keep in mind when working with context-dependent files.

Do Not Change System CDFs

Do not remove system CDFs, and do not change the way SAM sets them up. HP does not support system CDFs whose structure you have changed.

You can change the *contents* of configurable files, but do not remove CDF elements or change their names.

Create Few CDFs

In general, the fewer CDFs you create, the easier your system will be to administer.

Create Simple CDFs

Make your CDFs as simple in structure as possible.

Be wary of creating a CDF that:

- Contains elements (such as `remoteroot`) that match more than one cluster node's context.
- Contains some elements named for one type of context attribute (such as **cluster node type**) and other elements named for another type of attribute (such as **cluster node name**).

As the examples on the preceding pages show, creating complex CDFs can lead to results you may not expect when you are removing elements or adding new ones.

(See Chapter 2, "Understanding Clusters", under "What Is Context", for a list and explanation of context attributes, and for an explanation of the autocreation mechanism which comes into effect when you are adding elements.)

Be Careful when Moving CDFs or Creating Elements

As the examples in the previous section show, you need to think carefully when moving a CDF: you may inadvertently lose the hidden directory structure.

Similarly, you need to think about what result you want when moving or copying a regular HP-UX file so as to create a CDF element. See "Creating a New Element in a CDF" in the previous section.

Make sure you understand the autocreation mechanism, explained in Chapter 2, "Understanding Clusters".

Change Directories To See Inside a CDF

When you need to examine or modify a CDF, the most straightforward way to deal with the entire set of elements is to change your current working directory to the CDF's "hidden directory", for example:

```
cd /cdf.file+
```

Note the `+` sign, which acts as an escape character, allowing you access to the directory structure of a CDF. Now the `ls` command will show you all the elements of `/cdf.file`.

Administering Subsystems

System Accounting

`/usr/adm` is a CDF, to allow system accounting to be done for separately for each node in the cluster.

Mail

If your users will be sending mail to, or receiving it from, network nodes outside the cluster, you need to set up routing either through UUCP or ARPA Services.

- To configure UUCP, follow directions in the manual *Remote Access*.

  Configure UUCP on the cluster server.

- To configure mail via ARPA Services, follow directions for installing the **`sendmail`** utility in the manual *Installing and Administering ARPA Services*.

  Set up **`sendmail`** on the cluster server.

Line-Printer Spooling

A printer attached to any cluster node (the cluster server or any client) can be added to the line-printer spooler. Requests can be spooled from any cluster node, but the scheduler runs on the cluster root server only, even if spooled printers are attached to cluster clients.

Details are in Chapter 12, "Adding Peripherals to a Cluster".

UUCP

Although all UUCP lines must be on the cluster server, UUCP transfers can be initiated from any cluster node. UUCP is described in the manual *Remote Access*.

Cron

The `/usr/spool/cron` directory is a context-dependent file, allowing `cron` to run independently on each cluster node.

System V IPC

Messages, semaphores and shared memory are *not* distributed in a cluster.

Disk Quotas

Disk quotas are a means of controlling how much file system space a user can consume. You can limit both the number of files a user can keep, and the total number of 1-kilobyte blocks that he or she can consume. You set the limits by user and by file system. For each user, you set both a **soft limit** (a limit the user *should* not exceed) and a **hard limit** (a limit the user *cannot* exceed—file saves will fail).

When a user exceeds her soft limit, for files or blocks, the system sends her a warning. How the system reacts when a user exceeds a limit differs slightly between a cluster root server (or standalone computer) on one hand, and cluster clients on the other.

Users who exceed their limits will see the following behavior:

System Behavior When Disk Quotas Are Exceeded

Where Logged In	Limit Exceeded	System Response
Root Server	Soft limit for files	One warning whenever limit is exceeded.
	Soft limit for blocks	One warning whenever limit is exceeded.
	Hard limit for files	One warning when write is attempted.
	Hard limit for blocks	One warning when write is attempted.
Client	Soft limit for files	One warning whenever limit is exceeded.
	Soft limit for blocks	Warning when limit is exceeded and on each new write (up to maximum of one warning per minute) until usage falls below limit.
	Hard limit for files	One warning when write is attempted.
	Hard limit for blocks	May permit one write above limit.

Turning on Disk Quotas for a Locally Mounted File System

To turn on disk quotas on a **locally mounted file system** (a file system residing on a client's local disk), do the following:

1. Log in to the client to which the disk is attached.
2. Follow directions in Chapter 6, "Managing the File System", in the *System Administration Tasks* manual to set up disk quotas on this file system.

Some Special Cases

This section covers miscellaneous cases which may or may not apply to you. Read the first paragraph or two of each subsection to find out whether or not you need to read on.

System Files, CDFs, and Symbolic Links

CDFs are context-dependent files. You'll find a brief explanation and examples earlier in this chapter, and a more complete explanation in Chapter 2, "Understanding Clusters". **System files and directories** are those supplied on the HP-UX release tape or built during the `update` process.

Symbolic Links allow you to make one file name point to another, and are useful when you want to move a file or directory but continue to refer to it under the old name. (See `ln(1)` in the *HP-UX Reference*).

Read this section if one or more of the following apply:

- You plan to move and symbolically link an HP-UX system file or directory that is a CDF.
- You plan to move a system file or directory and symbolically link it to a path that contains CDFs.
- You have already done one or both of these or think that someone else has.

If any of the above applies to you, the advice that follows will help you avoid problems during your next system update. If you ignore the advice, the problems could be severe.

Rules and Guidelines

Remember that these rules apply to system files, and to subsystems and applications you install via `update(1M)`, but not to your own, "home-grown" files.

In general, avoid turning a CDF into a symbolic link or making a symbolic link point to a CDF. If you must do so, follow these rules.

Rule 1: Never Convert a CDF Itself to a Symbolic Link. Instead, convert the elements in the CDF to symbolic links.

See Chapter 2, "Understanding Clusters", and earlier sections of the present chapter, for information on the structure of CDFs.

Rule 2: Expand CDF Paths. If at all possible, avoid converting a file supplied by HP-UX into a symbolic link that points to a path containing CDFs; but if you must do this, then use the + character to expand all the CDF names in the path you use in creating the symbolic link.

If there is more than one CDF in the path you are pointing to, then *all* the CDFs in that path must be expanded.

See Chapter 2, "Understanding Clusters", and earlier sections of the present chapter, for examples of expanded path names.

Recovering Device Files from Old Back-ups

Read this only if one or more of the following apply:

- You depend on device files being "generic" (see below).
- You create device files using methods other than those recommended in *Installing Peripherals* and other HP-UX documentation.

By default, device files in a cluster are **cluster-node-specific**: each has a special field that identifies a particular cluster node (the server or a client) and only that node can use that file. This mechanism is analogous to, but not the same as, that used by **context-dependent files** (discussed earlier in this chapter). 8

You can see this special identifying field by using the -H option to ll, as in the following example.

```
# ll -H /dev+/client1
total 20
crw--w--w-   3 root     other      0 0x000000 client1   Dec  6 15:58 console
crw-rw-rw-   1 root     other     12 0x000000 client1   Dec  6 14:33 crt
crw-rw-rw-   1 root     other     19 0x150000 client1   Dec  6 14:33 ether
crw-rw-rw-   1 root     other     24 0x000010 client1   Dec  6 14:33 hil1
     :

# ll -H /dev+/client2
total 20
crw--w--w-   3 root     other      0 0x000000 client2    Dec  6 15:58 console
crw-rw-rw-   1 root     other     12 0x000000 client2    Dec  6 14:33 crt
crw-rw-rw-   1 root     other     19 0x150000 client2    Dec  6 14:33 ether
crw-rw-rw-   1 root     other     24 0x000010 client2    Dec  6 14:33 hil1
     :
```

But it is possible to create **generic** device files. These have a zero in the file name where the cluster node names (client1 and client2) are in the file names in the example, and they can be used by all cluster nodes.

If you recover generic device files from a backup, HP-UX will rename them as cluster-node-specific device files using the name of the cluster node on which you are performing the recovery.

This means, for example, that you can't use your cluster server to recover a device file intended for a cluster client; the file will get the server's node name and the client will not be able to use it. You will not be able to move or copy the file to give it the client's node name, but will have to re-create it by running mknod (1M), either with the cnode name option or on the client in question (cnode name defaults to the name of the current node).

(It may still be worth recovering the file, though, to see the major and minor number information which may otherwise be difficult to get.)

9

Backing Up Files in a Cluster

Backing up a cluster is very much like backing up a multi-user machine: you need to schedule backups regularly and at a time when users will not be unduly inconvenienced.

Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual, provides a complete introduction to system backup, and you should read that first if you are not used to doing backups.

This chapter concentrates on those aspects of the task that are specific to clusters, and gives some examples.

Guidelines for Cluster Backup

- Back up a cluster as often as you would back up a multi-user machine that was being used for similar purposes.

  Some administrators find that a schedule of weekly full backups and nightly incremental backups works well, but you must decide what is best for your system. The factors to consider are:

  - How *valuable* is the data (what is the cost of losing all or some of it)?
  - How *volatile* is the data (how often does it change)?

  Data that is both critical to your business and highly volatile may well need to be fully backed up more often than once a week, whereas data that is relatively stable or less important may not require more than a monthly full backup.

- Back up all the cluster's disks at the same time, whether they are attached to the server or to clients.

  An exception could be a locally mounted file system that contains a critical application and its data: in this case you might want to back up this file system more frequently than the cluster's other files.

- For efficiency, do system backups from the cluster root server.

  Again, there may be an exception if you are backing up *only* the files on a cluster client's local disk. In this case it could be more efficient to do the backup from the client in question. This is possible only if the client has (physically attached to it) its own tape drive or other backup medium; you must be logged in to the node attached to the backup medium when you issue the backup command.

- Use cluster options when backing up a cluster's files.

  The remainder of this chapter provides examples.

Cluster Backup: What's Different

- The main difference between backing up a cluster and backing up a standalone machine is that clusters use files, called context-dependent files or CDFs, whose contents differ depending on which member of the cluster is using them.

  These files are really directories, containing separate files (**elements**) for individual computers in the cluster. For this reason, context-dependent files are sometimes referred to as "hidden directories".

  The examples below show special options of standard HP-UX commands that will back up not only those elements that are specific to the node you are on (usually the cluster root server), but all other elements as well.

  Context-dependent files are explained in detail in Chapter 2, "Understanding Clusters".

- The other point to remember about cluster backup is that you must run the backup from the cluster node to which the backup device (tape drive, optical disk, etc.) is attached. (This does not apply of course to **network backups**—backups over the network to a storage device attached to a remote computer. See Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual, for information on network backups.)

Sample Backup Commands

This section contains examples of commands that will back up your cluster's files correctly, including the contents of "hidden directories" or context-dependent files.

If you are not used to doing system backups, or are planning to do a network backup, or need help in making a backup plan or deciding what kind of local storage device to use (DDS tape, cartridge tape, hard disk, optical disk), read Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual, before you go on.

Note

1. Remember that you must issue the backup command from the node to which the backup device is attached (unless you are doing a network backup).
2. Unless you have a good reason not to, you should do system backups from the cluster root server.

fbackup Command

When backing up a cluster's files with `fbackup`, use the `-H` (for "Hidden") option to back up the elements of context-dependent files which otherwise would not be accessible from the node you are on.

For more information on `fbackup(1M)`, refer to the *HP-UX Reference* and to "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual.

Full Backup Example

The following command does a full backup to DDS tape, on the basis of entries in the file `/usr/adm/fbackupfiles/graph` and using device file `/dev/rmt/0m`. It creates an index of the backed-up files in `/usr/adm/fbackupfiles/full0804.90`.

Note Type the command all on one line. (If the line is too long, let it run on: don't press (Return).)

```
fbackup -uH0f /dev/rmt/0m -g /usr/adm/fbackupfiles/graph
-I /usr/adm/fbackupfiles/full0804.90
```

Incremental Backup Example

The following command does an incremental (level 1) backup to DDS tape, on the basis of entries in the file `/usr/adm/fbackupfiles/graph` and using device file `/dev/rmt/0m`. It creates an index of the backed-up files in `/usr/adm/fbackupfiles/inc0809.90`.

Note Type the command all on one line. (If the line is too long, let it run on: don't press (Return).)

```
fbackup -uH1f /dev/rmt/0m -g /usr/adm/fbackupfiles/graph
-I /usr/adm/fbackupfiles/inc0809.90
```

tcio and cpio Commands

You need to use the `find(1)` command with the `-hidden` option when backing your system with `cpio(1)` and `tcio(1)`.

The following example backs up the entire file system to cartridge tape, using device file `/dev/update.src`.

```
cd /

find . -hidden -print | cpio -ocx | tcio -o /dev/update/src
```

SAM

You can use SAM (the System Administration Manager utility) to do cluster backups.

The procedure is the same as for a standalone machine, as described under "Backing Up Your System Using SAM" in Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual—with one exception: SAM allows you to choose whether or not to back up all the elements of context-dependent files.

Normally you will want to back up all elements, and in this case you don't need to do anything because the option is set to do this by default.

If you change the default, SAM will back up only those elements of context-dependent files that are specific to the cluster node you are logged in to (for example, if you are logged in to the cluster root server, SAM would back up only the server's version of `/etc/inittab`, not the clients'). If for some reason you want to do this, click on the option labelled `Back up all elements of context dependent files` under `Set Additional Parameters` in the `Add an Automated Backup` or `Backup Files Interactively` step menu.

Unless you are doing a **network backup** (backing up the cluster's files to a remote computer over the network) you must run SAM on the computer to which the backup device (tape drive, optical disk, etc.) is physically attached; normally this should be the cluster root server.

For information on network backups, see Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual.

Recovery

You do not need any special options to recover a cluster's files from a backup.

For example, to recover a cluster's entire file system backed up to DDS tape with the `fbackup(1M)` utility, you would use a command like this:

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

You can also use SAM to recover backed-up files, as described in Chapter 8, "Backing Up and Restoring Your Data", in the *System Administration Tasks* manual.

10

Booting and Shutting Down Clusters and Cluster Nodes

This chapter explains how to boot and shut down a cluster, and how to boot and shut down individual nodes.

Cluster clients boot over the LAN from kernels stored in the cluster server's disk space. Booting a client for the first time is described in Chapter 5, "Adding Cluster Clients". After that, booting cluster nodes is fairly simple; see "Booting the Server" and "Booting a Cluster Client", later in this chapter.

When you shut down cluster nodes, there are four different cases that you need to recognize and handle appropriately:

- Shutting down the cluster server (root server).

  This shuts down the entire cluster. See "Shutting Down a Cluster", later in this chapter.

- Shutting down a cluster client that has no local disk, or has a disk which is being used only for that client's local swap.

  This is the simplest case.

  See "Shutting Down a Simple Client", later in this chapter.

- Shutting down an **auxiliary file server.**

  An auxiliary file server is a client whose local disk space is being used for file space. When you shut down an auxiliary file server, you automatically unmount the file system on the local disk. See "Shutting Down an Auxiliary File Server", later in this chapter, which also covers the case of a client whose local disk is being used both for a file system and **distributed swap** (that is, swap space which is being used by other clients).

- Shutting down an **auxiliary swap server.**

  An auxiliary swap server is a client whose local disk space is being used by other clients for swap space (distributed swap space). Shutting down an auxiliary swap server shuts down the clients that are swapping to this client's disk space. See "Shutting Down an Auxiliary Swap Server", later in this chapter.

Booting a Cluster

To boot a cluster you first boot the root server, then boot the clients.

Booting the Root Server

Boot the root server just as you would a standalone machine.

For details, see Chapter 3, "Starting and Stopping HP-UX" in the *System Administration Tasks* manual, HP part number B1864-90010.

Booting a Cluster Client

Before you boot a cluster client, check that all the following conditions are true:

- You have added this computer to the cluster.

  See Chapter 5, "Adding Cluster Clients".

- The cluster server has completed its boot sequence and is in multi-user mode.

  You can check by entering the command:

  ```
  who -r
  ```

  You should see something like this:

  ```
        .       system boot   Apr   5 09:33    2    0    S
  ```

  The column that matters is the third from the right, which contains the numeral 2 in the above example, indicating multi-user mode.

  If your root server is in single-user mode, you'll see something like this:

  ```
        .       run-level S   Apr   5 13:58    S    1    2
  ```

  To change from single-user to multi-user mode, enter the command:

  ```
  init 2
  ```

- If the client you are booting will swap to another client (a **swap server**), be sure that the swap server has completed its boot sequence and is in multi-user mode.

- The client is part of only one cluster.

  This means that only one cluster root server on the LAN has an entry in its `/etc/clusterconf` file for this cluster client.

  This is not a requirement (see below), but it makes things simpler.

- No disk attached to the client that you are booting contains a bootable system.

  This is not a requirement (see below), but it makes things simpler.

Once you've made sure the client meets all these conditions, you can begin the boot process.

1. Turn on power to the computer.
2. Allow the system to boot in **unattended mode** (that is, do nothing until the boot process is complete).

When the cluster client displays a login prompt, it is ready for use.

Cluster Client Doubling as Standalone System

If the cluster client will double as a standalone workstation, and therefore has a bootable system on a disk that is attached to it, you must always boot the client in **attended mode** and choose the operating system you need (as described under "Preparing To Add a Series 300/400 Client" in Chapter 5, "Adding Cluster Clients").

Setting up a client this way will complicate the task of administering the cluster: don't do it unless you have a good reason to.

Cluster Client Belonging to More than One Cluster

If the cluster client will belong to more than one cluster (that is, has an entry in more than one cluster server's `/etc/clusterconf`) you must always boot the client in **attended mode** and choose the operating system you need (as described under "Preparing To Add a Series 300/400 Client" in Chapter 5, "Adding Cluster Clients").

Setting up a client this way will complicate the task of administering the cluster: don't do it unless you have a good reason to.

Shutting Down a Cluster

Caution Shutting down the cluster root server shuts down the entire cluster.

To shut down the cluster:

1. Log in to the *cluster root server* as superuser.
2. Be sure that you are in the root directory by entering the command:

```
cd /
```

3. Shut down the system using the command:

```
/etc/shutdown -h
```

Note You do not have to shut down the cluster clients individually before shutting down the cluster server. If clients are still up and running when you issue the `shutdown` command on the server, the server's console will display warnings to that effect and the system will wait an additional grace period to allow client users time to log out. If you want the server to shut down as quickly as possible, shut down the clients first.

The remainder of this chapter deals with shutting down cluster clients.

Shutting Down a Cluster Client

Caution

How you shut down a cluster client depends on whether or not the client has a local disk, and how that disk is being used.

- If the cluster client has no local disk, follow directions under "Shutting Down a Simple Client", later in this chapter.
- If the cluster client has a local disk which it is using for swap, and no other client is swapping to that disk, follow directions under "Shutting Down a Simple Client", later in this chapter.
- If the cluster client has a local disk which is being used for a file system, or for a combination of swap and a file system, follow directions under "Shutting Down an Auxiliary File Server", later in this chapter.
- If the cluster client has a local disk which other clients are using for swap, but there is no file system on the disk, follow directions under "Shutting Down an Auxiliary Swap Server", later in this chapter.

Allowing Users Shutdown Capabilities

A user does not need to be superuser to halt or reboot a cluster node. You can use the `/etc/shutdown.allow` file to give permission for specific users to shut down specific computers in the cluster.

What you will probably want to do is to allow the "owners" of workstations in the cluster to shut down their own local nodes. You give this permission by entering the user login name and the cluster node name in the file `/etc/shutdown.allow`. For example, to allow user `fred` to shut down `client1`, make the following entry in `/etc/shutdown.allow`:

```
client1 fred
```

The superuser, and possibly other users, will need to be able to shut down all the cluster nodes, and you may want to allow some cluster nodes to be shut down by anyone. You can use a wildcard character in such cases; for example, the following entry in `/etc/shutdown.allow` allows the superuser to shut down all the cluster nodes:

```
+ root
```

For more information, see `shutdown(1M)` in the *HP-UX Reference*.

Caution

Be careful when adding entries to `/etc/shutdown.allow`. This is how it works:

- If `/etc/shutdown.allow` does not exist, or exists and is empty (the default), then the superuser, and only the superuser, can shut down any cluster node.
- If `/etc/shutdown.allow` is not empty, then only the users listed in it can execute the **shutdown** command, and they can shut down only those systems listed beside their login names.

  If you use `/etc/shutdown.allow`, you must make sure that it contains all the permissions you need to grant, *including the superuser login and the systems the superuser can shut down.*
- An entry in `/etc/shutdown.allow` allows an ordinary user to halt or reboot the system named, but not to bring it down to single-user state. This capability is reserved for the superuser.

Shutting Down a Simple Client

The term **simple client** refers to a client that you can shut down without affecting any other cluster nodes. A simple client

- Has no local disk

  or

- Has a local disk which is being used only for swap, and only by that client.

Shut down a simple client as follows:

1. Log in to the *cluster client*.

   You need not log in as superuser to shut down a cluster client if your login name is matched with this system in `/etc/shutdown.allow`. See "Allowing Users Shutdown Capabilities", earlier in this chapter.

2. Change directories to the root directory using the command:

   ```
   cd /
   ```

3. Make sure that any other users of this cluster client have logged off.

4. Shut down the system:

   ```
   shutdown -h
   ```

Shutting Down an Auxiliary File Server

An **auxiliary file server** is a cluster client that has a disk attached to it that contains a file system.

Shutting down an auxiliary file server will automatically unmount any file system on the local disk(s).

Use the procedure that follows to shut down a client whose local disk space is being used for:

- A locally mounted file system

  or

- A locally mounted file system and distributed swap space (swap space used by other clients).

Caution

1. Make sure that users of clients with local disks do not shut down their computers unless it's safe to unmount any locally mounted file systems.
2. If the local disk or disks also contain swap space that is being used by other clients, shutting down the client that has the disk will also shut down the clients that are swapping to it. Users of clients with local disks must not shut down their computers until it's safe to shut down all other clients that are swapping to the local disk space.

Shut down an auxiliary file server as follows:

Do this on the client that has the disk attached to it.

1. Log in to the client with the local disk.

Note

You need not log in as superuser to shut down a cluster client if your login name is matched with this system in `/etc/shutdown.allow`. See "Allowing Users Shutdown Capabilities", earlier in this chapter.

You do need to be superuser, however, to override `msgno` when warning users to vacate locally mounted file systems, as shown in the next step. (`Msgno` prevents messages being written to the screen.)

---

2. Get all users out of the file system(s) in question.

   For example:

   ```
   /etc/cwall

   FILE SYSTEM /users2 ABOUT TO BE UNMOUNTED!

   IF YOU'RE WORKING ANYWHERE UNDER /users2,
   SAVE YOUR FILES AND CHANGE DIRECTORIES NOW!!

   [CTRL] D
   ```

   A file system can't be unmounted as long as anyone (or any program) has their working directory in any branch of the file system.

   If you need to check which file systems are mounted locally, use `mount -l`,

   ```
   /etc/mount -l
   ```

   You should see one or more entries like this:

   ```
   /users2 on /dev/dsk/c1d0s8 read/write on Thu Apr  5 09:11:53 1990
   ```

   This shows locally mounted file system(s) and the associated device file(s). In this example, the file system `/users2` is associated with device file `/dev/dsk/c1d0s8`.

(This option of `mount(1M)` does not show file systems mounted via NFS mount, and those file systems will not automatically be unmounted when the client is shut down. `/etc/mount - L` shows all file systems that *can* be unmounted from this client, including file systems mounted via NFS mount.)

Caution If your local disk is also being used as swap space by other clients, shutting down your client will also shut down the clients that are swapping to your disk. Warn the users of those clients too.

---

3. Shut down the cluster client:

```
/etc/shutdown -h
```

Shutting Down an Auxiliary Swap Server

An **auxiliary swap server** is a cluster client that has a local disk to which other members of the cluster are swapping. (We use the term **distributed swap** to describe this kind of swapping.) When you shut down an auxiliary swap server, you shut down all the cluster clients that are swapping to its disk space.

If you're shutting down a client whose local disk is being used for a locally mounted mounted file system as well as distributed swap, follow the procedure under "Shutting Down an Auxiliary File Server", earlier in this chapter, instead.

To shut down an auxiliary swap server, do the following.

Caution Shutting down an auxiliary swap server shuts down all cluster clients that are swapping to this auxiliary server's local disk(s).

Do this on the client that has the local disk.

1. Log in to the client that has the local disk.

Note You need not log in as superuser to shut down a cluster client if your login name is matched with this system in `/etc/shutdown.allow`. See "Allowing Users Shutdown Capabilities", earlier in this chapter.

---

2. Shut down the auxiliary server:

```
/etc/shutdown -h
```

11

Reconfiguring the Kernel for a Cluster Node

Chapter 2, "Constructing an HP-UX System", in the *System Administration Tasks* manual, explains in general when you need to reconfigure an HP-UX kernel, and how to do it. If you are new to system administration, or not used to reconfiguring the kernel, you may want to stop and look through that chapter now to get an idea of what's involved.

This chapter deals with those aspects of kernel configuration that are unique to clusters: things you must do differently if you're modifying a cluster node's kernel as opposed to a standalone computer's kernel.

How Kernel Files Are Stored in a Cluster

Each member of a cluster (the server and each cluster client) has its own kernel, which is stored as an element of the context-dependent file /hp-ux+. For example, if a cluster consists of three computers named server, client1, and client2, /hp-ux+ will have the following structure:

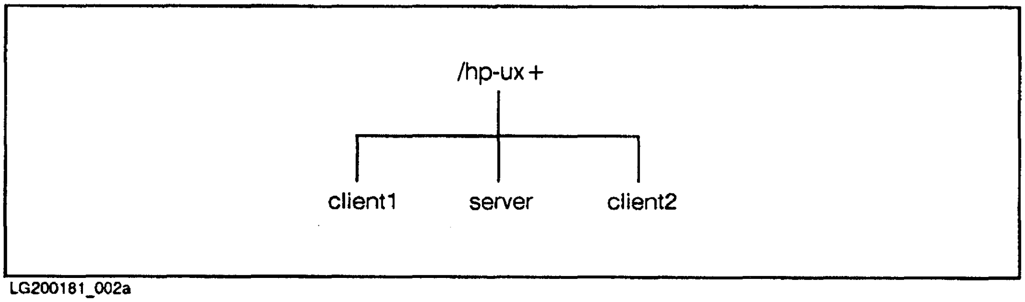


Figure 11-1. Structure of Kernel File /hp-ux+

This means that if a client's node name is client1, the full pathname of its kernel file will be /hp-ux+/client1, though when you're logged in to client1 you would normally access this file simply as /hp-ux.

Context-dependent files are explained in detail in Chapter 2, "Understanding Clusters", in this manual.

When Do You need To Modify The Kernel?

You may need to modify the cluster server's kernel for any of the reasons described in Chapter 2, "Constructing an HP-UX System", in the *System Administration Tasks* manual, but if you have to modify a client's kernel, it will probably be for one of these reasons:

- To configure a local disk for swap and/or a file system.

  Read the "Rules and Guidelines" below, then follow the procedure for adding a local disk in Chapter 12, "Adding Peripherals to a Cluster", in this manual.

- To configure other local peripherals such as a tape drive or printer.

  Read the "Rules and Guidelines" below, and the notes and examples for tape drives or printers and plotters in Chapter 12, "Adding Peripherals to a Cluster", in this manual.

  Chapter 9, "Managing Printers and Printer Output", in the *System Administration Tasks* manual, has cookbook procedures for adding printers and plotters. If you're already familiar with the task of adding a local printer or plotter to *a cluster client* and understand how spooling works in a cluster, you can probably go straight from here to those procedures; but don't skip the material in the present manual if you haven't already read it.

- To add or delete an HP-UX subsystem.

Caution If you are adding or deleting NFS or CDFS (CD-ROM file systems), follow directions under "Adding and Deleting CDFS and NFS" later in this chapter.

- □ The normal way to add HP-UX subsystems is to use the `update(1M)` utility: see *Installing and Updating HP-UX*.
- □ If you *only* need to configure the subsystem into the kernel (that is, if the subsystem software itself is already on your system and is compatible with the current kernel), you can use SAM.

  Even in this case, however, we recommend that whenever possible you use `update` and the update source (tape, disk or network server) from which you last updated the system. This will ensure that the software and the kernel are compatible. `Update` will reconfigure the kernel for you in the process of adding the software.
- □ To remove a subsystem from the kernel, use SAM, following directions in Chapter 2. "Constructing an HP-UX System", in the *System Administration Tasks* manual.

You may also occasionally want to modify a client's kernel parameters. There's a simple example later in this chapter.

Rules And Guidelines

The procedures for modifying the kernel of a cluster server or cluster client are the same as for a standalone machine, so long as you follow these rules:

- Make sure you are logged in (directly or remotely) to the cluster node whose kernel is to be modified.

  Otherwise you will modify the wrong kernel.

- *Never copy a client's kernel to /SYSBCKUP!*

  Otherwise you will overwrite the backup copy of the *cluster server's* kernel. This is because `/SYSBCKUP` is not a context-dependent file, and by HP-UX convention it is reserved for the cluster server.

  To make a copy of a client's kernel, copy it to a name unique to the client. For example, for a cluster client named `client1` you might call the backup kernel `/sysbckup.client1`.

Note When you use SAM to do a task that involves reconfiguring the cluster server's kernel, SAM copies the old kernel to `/SYSBCKUP` for you; but on a client, SAM does not copy the old kernel at all, so you need to do this yourself before running SAM.

- Install one kernel before modifying the next.

  When you need to modify more than one cluster node's kernel, install each new kernel in `/hp-ux` before you start to modify the next.

  You need to do this because the working files output by the HP-UX utilities that build the kernel are not context-dependent files: they are overwritten each time you reconfigure any cluster node's kernel. For example, `/etc/conf/hp-ux` is overwritten each time you modify any cluster node's kernel. This happens whether or not you use SAM.

  If you're using SAM, SAM will offer to install the new kernel for you; make sure you accept this option before leaving SAM if you're about to modify another cluster node's kernel.

- Use `/etc/conf/dfile` as the source file for regenerating the kernel.

  In a cluster, `/etc/conf/dfile` is a context-dependent file with the same structure as `/hp-ux` (shown earlier in this chapter): there is a separate element for each member of the cluster. This ensures that when you edit `/etc/conf/dfile` on one node, you don't overwrite another node's `dfile`.

  If for some reason you need to use some other name for your kernel source file, make sure you do two things:

  - □ Give the source file a name that uniquely identifies this cluster node, for example `dfile.client1`.
  - □ Copy this other source file to `/etc/conf/dfile` as soon as possible after generating the kernel.

Caution On all systems, whether or not they belong to a cluster, `/etc/conf/dfile` should match the running kernel; system software depends on it.

  When you use SAM to modify a cluster node's kernel, SAM, by default, overwrites that node's version `/etc/conf/dfile` so that it reflects the new kernel. See "Example: Changing Kernel Parameters on a Cluster Cluster Client", later in this chapter, for details.

- Beware of CDFS (CD-ROM file systems) and NFS!

  Don't boot a cluster node (server or client) from a kernel that contains CDFS or NFS unless all the currently running cluster kernels have it; conversely, don't boot a cluster node from a kernel that does *not* have CDFS or NFS unless all the other kernels don't have it.

  See the next section, "Adding and Deleting CDFS and NFS".

Adding and Deleting CDFS and NFS

The following subsystems must be configured into *all or none* of the cluster nodes' kernels (into the server's kernel and *every* client's kernel or else into *none* of the kernels):

- NFS
- CDFS (the subsystem "driver" for CD-ROM file systems)

Caution If one of these subsystems is configured into the server's kernel, but not into one or more of the clients', then the affected clients will panic when you try to boot them.

Proceed as follows.

- To add NFS to the cluster's software, use the HP-UX utility `update(1M)`. `Update` will load the software and configure the cluster node's kernels correctly.

  See Chapter 14, "Updating a Cluster", in this manual, and *Installing and Updating HP-UX*, for more information.
- To remove NFS from the kernel, or to add or delete CDFS, use SAM, following the procedure on the next page.

To configure CDFS into the cluster, or remove CDFS or NFS from the cluster, do the following.

1. Make a backup copy of each cluster client's kernel.
   a. Log in to the *client* as superuser.
   b. Copy `/hp-ux` to a backup file (*not* `/SYSBCKUP`). For example:

   `cp /hp-ux /sysbckup.client1` *(Use a name that will identify this client)*

2. Reconfigure the root server's kernel.

   When you have backed up each client's kernel, do the following:

   a. Log in to the *root server* as superuser.
   b. Use SAM to configure the subsystem into the root server's kernel, or remove the subsystem from the kernel (as appropriate).
   c. Have SAM regenerate and install the new kernel.

3. Have SAM reconfigure the clients' kernels.

   SAM will now offer to run a script on each cluster client to generate and install new kernels, compatible with the root server's. Proceed as follows:

   a. Make sure all the clients are booted to the cluster.
   b. Respond [Yes] to have SAM regenerate the clients' kernels and install them in `/hp-ux`.
   c. Wait for all the scripts running on the clients to complete.

      SAM will keep you informed about the status of the scripts running on the clients.

      When a script completes on a given client, the new kernel will be installed in that client's version of `/hp-ux` (see "How Kernel Files Are Stored in a Cluster", at the beginning of this chapter), and the client's version of `/etc/conf/dfile` will reflect the change.

d. Reboot the server.

SAM will do this for you, or you can get out of SAM and do it yourself if you prefer. In either case, SAM will install the server's new kernel in `/hp-ux`.

The clients will reboot from their new kernels.

Example: Changing Kernel Parameters on a Cluster Client

You may want to use the following example as a tutorial to help you get used to the kernel portion of SAM and to find out how SAM uses and changes the kernel files.

Suppose you want to increase the number of lines of simulated terminal memory available on your cluster client's console. The parameter in question is `scroll_lines`. (This parameter does not affect X Windows, only the **Internal Terminal Emulator (ITE)** which is in operation, for example, on the system console when you first boot the workstation.)

To change `scroll_lines` from the default, 100 lines, to 200 lines, follow these steps.

Caution Do this on the cluster client whose kernel you are going to modify.

1. Copy `/hp-ux` to a backup file (*not* `/SYSBCKUP`). For example:

   `cp /hp-ux /sysbckup.client1` *(Use a name that will identify this client)*

Note `/SYSBCKUP` is reserved for the cluster server's backup kernel.

2. Run SAM.

   Log in as superuser and enter:

   `/usr/bin/sam`

   If you have not used SAM before, the tutorial in Chapter 1, "Introduction to System Administration", in the *System Administration Tasks* manual, will help get you acquainted with it.

3. Select:

   `Kernel Configuration`

   then

   `Configurable Parameters`

   You'll see a screen that allows you to change any or all of the configurable parameters.

   System parameters and their meanings are documented in Appendix A, "System Parameters", in the *System Administration Tasks* manual.

4. Change the value for `scroll_lines` to 200:

   a. Select the `scroll lines` parameter from the list.
   b. Pull down the `Actions` menu and select `Modify Configurable Parameter...`.
   c. Put the cursor in the text edit field and type in 200.
   d. Press the (OK) button.

Caution Before you go on to the next step, make sure all other users of this cluster client are logged off.

If this is an **auxiliary swap server**, rebooting it will also reboot the other clients swapping to its local disk space.

If this is an **auxiliary file server**, you will not be able to reboot it if any users or programs have files open in the locally mounted file systems.

See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", for details.

5. Let SAM rebuild and install the kernel and reboot the system for you:
   a. Pull down the `Actions` menu and select `Create a New Kernel`.
   b. Choose (Yes) in the confirmation box.
   c. Select `Create a new kernel now`.
   d. Select `Move the kernel into place and reboot the system now`.
   e. Press the (OK) button.

SAM now does the following:

1. Calls `config` and `make` to build a new kernel.
2. Installs the new kernel.

   SAM moves the new kernel into the context-dependent file `/hp-ux`, as `/hp-ux+/`*cluster_nodename*.

   For example, on a cluster node named `client1`, the SAM writes out a new kernel file whose full pathname is `/hp-ux+/client1`.
3. Saves the corresponding `dfile` (by default) as `/etc/conf/dfile`.

   This too is a context-dependent file; on `client1`, its full pathname would be `/etc/conf/dfile+/client1`.

Note If you opt not to have SAM overwrite your existing `/etc/conf/dfile`, SAM saves the `dfile` as `/etc/conf/dfile.SAM` instead.

SAM lets you save the `dfile` under this alternative name in case the old `/etc/conf/dfile` contains comments you want to keep. If so, you should copy the comments into `/etc/conf/dfile.SAM` and move the resulting file to `/etc/conf/dfile` as soon as possible because other HP-UX software expects `/etc/conf/dfile` to reflect the running kernel.

`/etc/conf/dfile.SAM` is *not* a context-dependent file, and will be overwritten next time you reconfigure the kernel for any cluster node.

---

4. Reboots the cluster client, activating the new kernel.

If the New Kernel Doesn't Boot

If the boot fails after you have regenerated the kernel, you can boot from a backup kernel. How you do this depends on whether the kernel belongs to the cluster root server or a cluster client. See "Booting the Root Server from the Backup Kernel" and "Booting a Cluster Client from the Backup Kernel", below.

After booting, you should try to diagnose what went wrong. Use *Solving HP-UX Problems*, HP part number B2355-90030 to assist you. (See Chapter 4, "Diskless Cluster Problems", and Chapter 5, "System Boot-Up Problems", in that manual)

Booting the Root Server from the Backup Kernel

Boot a cluster server from the backup kernel (normally `/SYSBCKUP`; it will definitely be `/SYSBCKUP` if you used SAM to generate and install the kernel). You'll find directions in Chapter 2, "Constructing an HP-UX System", in the *System Administration Tasks* manual.

Once the root server has rebooted, all the clients should come back up automatically.

Booting a Cluster Client from the Backup Kernel

If the boot fails when you are trying to update a cluster client's kernel, do not boot using `/SYSBCKUP`. `/SYSBCKUP` contains the server's backup kernel.

Instead, log in to the cluster root server and move the file you saved as the backup kernel (for example `/sysbckup.client1`) to the client's element of the context-dependent file `/hp-ux`.

For example, given a client named `client1` and a backup kernel file named `/sysbckup.client1`, you'd enter

```
mv /sysbckup.client1 /hp-ux+/client1
```

This will replace the kernel that won't boot with the backup kernel, and now the client should boot.

(For an explanation of context-dependent files, see Chapter 2, "Understanding Clusters".)

12

Adding Peripherals To A Cluster

This chapter explains how to add peripherals to the server and clients in a cluster. It covers the options and restrictions, and their implications, and provides some examples of adding peripherals.

"Peripherals" in this chapter refers specifically to the following:

1. Disk drives
2. Tape drives
3. Printers/plotters
4. Modems
5. Terminals

Depending on the peripheral itself, and how and where you add it in the cluster, a peripheral may be available to all users of the cluster, or may be restricted to users logged in to the cluster node it's attached to. This means that you should plan carefully before adding a given piece of hardware to a particular node.

Use Table 12-1 as a starting point, then read the part of this section that explains the rules for the peripheral you're interested in, then go on to "Adding Peripherals to the Cluster Root Server" or "Adding Peripherals to a Cluster Client" for procedures, examples, and some further advice.

You will also need to read the section dealing with the particular model of peripheral you want to add in the *Installing Peripherals* manual.

How Clusters Support Different Peripherals

An HP-UX cluster combines the advantages of a multi-user system and a workstation, providing each user access to a pool of I/O and storage resources, while reserving the power of each workstation's processor to one person (or to a small group of terminal users).

In most cases, you can attach a given peripheral either to the root server or to any cluster client. Depending on the type of peripheral, its resources will be:

- automatically available to all members of the cluster (as with a file system on a disk);

  or

- exclusively available to the cluster node the peripheral is attached to (as with a tape drive);

  or

- configurable either way (as with swap).

  The remainder of this section explains the choices and the rules for attaching peripherals to cluster nodes.

Shared vs. Exclusive Resources

Table 12-1 shows the choices you have for distributing peripheral resources in a cluster. In this table, the "Shared?" column shows whether the resource can be shared with other nodes in the cluster, while the "Exclusive?" column shows whether the resource can be privately owned by the cluster node it's attached to. A "yes" in both columns means that the resource (a printer for example) can be configured to be either shared or exclusive.

12

Table 12-1. Availability of Resources in a Cluster

Resource	Shared?	Exclusive?	Comments
Disk drives:			
file systems	Yes	No	Always shared, no matter where disk drive is attached. Root must be on server's disk.
device swap	Yes	Yes	Server's swap usually shared. Client's swap on local disk can be exclusive or shared.
file system swap	Yes	Yes	Root server and clients can enable file system swap only to their own disks.
Tape drives	No	Yes	Available only to users logged in to the node where the drive is attached.
Printers, plotters	Yes	Yes	Spooled printers can be attached to any node and are available to all. Unspooled printers available only to local node.
Modems	No	Yes	Available only to users logged in to the node where the modem is attached. Modems used with **cu** and **uucp** supported only on root server.
Terminals	No	Yes	Users log in on local node. Can **rlogin** to other nodes.

The following pages explain the rules for each type of peripheral.

Disk Drives

You can attach disk drives to the cluster server and to any client. Whether the space on a given disk is shared or not depends on how you define that space.

As with any disk under HP-UX, disk space on a drive attached to a cluster client can be used for file system(s), or swap space, or a combination of the two.

Swap Space

The rules governing swap in a cluster are:

- You must configure a swap area on a disk attached to the cluster root server.
  - The server must swap to this swap area.
  - Any or all clients may also swap to this swap area.
- You can configure additional swap space on any disk attached to any client.
  - Other clients can swap to this swap area.

    A client to whose disk other clients are swapping is called an **auxiliary swap server** or **swap server**. Shutting down a swap server shuts down all the clients that are swapping to its disk space. (For more information on this, see Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes").

    We'll call this kind of swap **distributed swap** when we need to distinguish it from **shared swap** on the root server's disk.
  - A swap server must swap to its own local disk.
  - The cluster root server cannot swap to a client's local disk.
- Each cluster node must swap to only one node's disk space.

  For example, a client cannot have its primary swap area on the server's disk and secondary swap on its own or another client's disk.

- File system swap can be configured on the root server's disk and on any client's disk.
  - ▫ A client must be configured to swap to its own disk space before you can configure file system swap in that disk space.
  - ▫ You must be logged in to the node that has the disk (root server or auxiliary server) to enable swap to a file system that physically resides on that disk.

    For example, in the cluster shown in Figure 12-1, a little later in this chapter, `server` cannot enable swapping to `/users/fred` or `/users/joe`.
  - ▫ Once a root server or auxiliary server has enabled swapping to a file system, all nodes swapping to this root or auxiliary server will swap to that file system.

    For example, in the cluster shown in Figure 12-1, a little later in this chapter, if `client1` is a swap server for `client2`, and `client1` enables swapping to the locally mounted file system `/users/fred`, then both `client1` and `client2` will start swapping to `/users/fred`.

    See "Setting up Swap to an Auxiliary Server", under "Local Disks" later in this chapter, for information on setting up swap to another client's local disk space. Directions for setting up file system swap are in Chapter 7, "Managing Swap Space", in the *System Administration Tasks* manual.

File system swap in a cluster conforms to the same rules as device swap: each node still swaps to only one node's disk space, and root and auxiliary servers swap to their own disk space.

Summary.

- The cluster root server must always swap to its own disk space.
- Auxiliary swap servers must also swap to their own disk space.
- A cluster client that is not an auxiliary swap server can swap to any one of the following:
  - ▫ Its own local disk space.
  - ▫ The cluster root server's disk space.
  - ▫ Another client's disk space (if that client is swapping there).

File Space

All file systems are shared in a cluster. A client with a local disk can, if you so decide, keep exclusive access to *swap space* on the disk, but any *file system* you configure on the disk must be mounted to the cluster's global file system tree (whose root must be on the cluster root server's root disk) and as such is automatically available to all the other nodes. A cluster client, like the cluster root server, cannot have its own "private" file system.

A cluster client with a file system on its local disk is called an **auxiliary file server**, and the file system is called a **locally mounted** file system.

Note The above, and the discussion that follows, apply only to disk drives that are configured into the cluster.

It is possible for a cluster client to have disks that are *not* configured into the cluster; for example a client doubling as a standalone workstation will have a disk or disks that contain a bootable system, and that system and the files it contains will not be available to the cluster. (For instructions on booting such a client, see Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".)

Locally Mounted File Systems: Rules. The rules for **locally mounted file systems** (file systems installed on a client's local disk) are:

- The root file system (`/`) must be on the cluster root server's disk.
- Only HFS file systems are locally mountable.
  - CDFS (CD-ROM) file systems must reside on a drive attached to the cluster root server.

12

Caution You must, however, configure the CDFS software "driver" into *every* cluster node's kernel. See Chapter 11, "Reconfiguring the Kernel for a Cluster Node".

- The mount point for an NFS mount must be a directory that resides on the cluster root server's disks.
  - You *cannot* do an NFS mount onto a locally mounted file system.
  - A locally mounted file system *can* itself be NFS mounted (exported).

In Figure 12-1, a little later in this section, `/users/fred` could not be the mount point for an NFS mount (that is, a file system from a remote system could not be attached to this cluster's file system tree at `/users/fred`).

The file system `/users/fred` could, however, be NFS mounted onto a remote system.

Caution If you install NFS, you must configure the NFS software into *every* cluster node's kernel. See Chapter 11, "Reconfiguring the Kernel for a Cluster Node".

- For HFS file systems:
  - The mount point for a locally mounted file system must be a directory residing on a device that is attached *either* to the cluster root server, *or* to the same cluster client: it cannot be on another client's disk.
  - Similarly, the mount point for a file system residing on a disk attached to the cluster root server must also be on a disk attached to the cluster root server: it cannot be on a client's disk.

  See Figure 12-1, a little later in this section.
- Locally mounted file systems are automatically visible to all members of the cluster; a cluster client cannot have its own "private" file system.
- An auxiliary swap server can enable file system swap on its own locally mounted file system, but not on another client's locally mounted file system.
- You must execute `mount(1M)` (except for purely reporting purposes) on the cluster node to which the local disk is attached. This applies to other file system commands as well.

  The commands in question are:
  - `fsck(1M)`
  - `fsclean(1M)`
  - `fsdb(1M)`
  - `fuser(1M)`
  - `mediainit(1)` (for a disk)
  - `mkfs(1M)`
  - `mount (1M)` (except for reporting purposes)
  - `newfs(1M)`
  - `tunefs(1M)`
  - `umount(1M)`
  - `mkrs(1M)` (Series 300/400 only)

  The numbers in parentheses after the names of the commands refer to the section in the *HP-UX Reference* where you can find out more about the command.

- Keep HP-UX system software on the cluster root server's disk.

  Don't move HP-UX system directories such as `/usr` and `/bin` to a client's local disk; otherwise you may run into trouble next time you run `update`. For more information see Chapter 14, "Updating a Cluster".

12

Locally Mounted File Systems: Example.

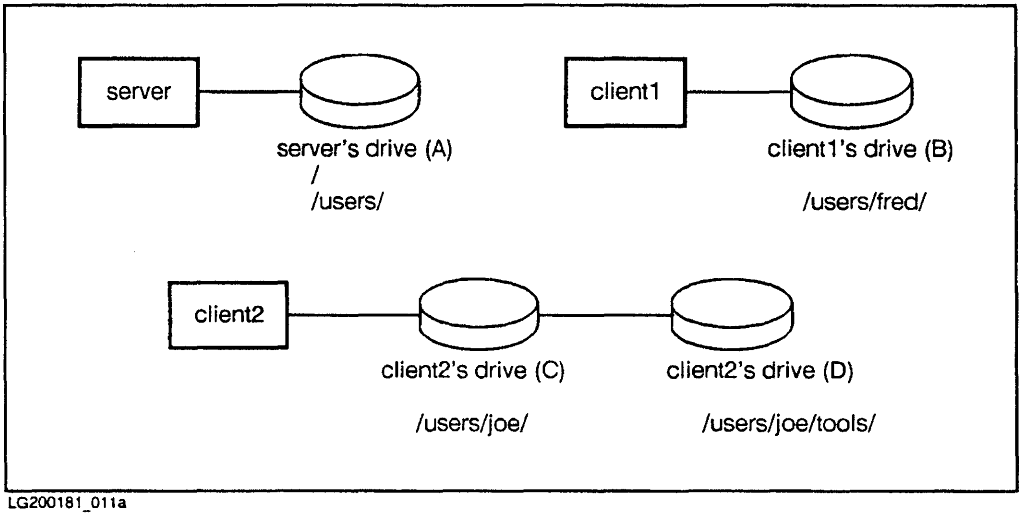


Figure 12-1. Locally Mounted File Systems in a Cluster

In this example, the cluster root server (`server`) and two clients (`client1` and `client2`).

- `server` has one disk drive (`A`).

  Given that `server` has only this one disk, then:

  - Any NFS mounts to the cluster's file system must be made to directories on disk `A`.
  - The mount point for the first file system mounted locally by `client1` and `client2` must be a directory on disk `A`.

- `client1` has one disk drive (`B`).

  `B` contains the file system `/users/fred`, mounted on `/users`, which is on the cluster root server's disk (`A`).

- `client2` has two disk drives, `C` and `D`.

  - Drive `C` contains the file system `/users/joe`, which, like `/users/fred`, is mounted on`/users` on the cluster root server's disk `A`.
  - Drive `D` contains the file system `/users/joe/tools`, which is mounted on `/users/joe` on `client2`'s disk `C`.

    This follows the rule that the mount point must be either on the cluster root server's disk, or, as in this case, on a disk attached to the same client.

- The following are examples of mounts that are illegal given the configuration shown above:

  - `/users/fred` cannot be a mount point for any file system residing on disk `A`, `C` or `D`.
  - `/users/joe` and `/users/joe/tools` cannot be mount points for any file system residing on disk `A` or `B`.

For More Information

- You add a disk drive to the cluster root server just as you would to a standalone machine. Use the chapter on installing disk and tape drives in the *Installing Peripherals* manual.
- For information on installing and using a disk drive on a cluster client, see "Local Disks" under "Adding Peripherals to a Cluster Client", later in this chapter.

Tape Drives

The rules for tape drives in a cluster are:

- You can connect tape drives to the cluster server and to any cluster client.
- To use a tape drive, you must be logged in (directly or remotely) to the node to which the drive is attached. 12

The following are points to keep in mind:

- Back up the cluster's files to a tape drive attached to the cluster root server.

  This will be quicker than backing up to a client's local tape drive.

  See Chapter 9, "Backing Up Files in a Cluster", for backup examples.
- You must use a tape drive attached to the cluster root server whenever you use the `update(1M)` utility to install or update software.

  See Chapter 14, "Updating a Cluster" for more information.

Printers, Plotters

The rules for printers and plotters are:

- Printers and plotters can be attached to the cluster server and to any cluster client.
- The line-printer spooler runs only on the root server.
- Any printer or plotter, attached to any node, can be shared by all members of the cluster (by means of the line-printer spooler running on the root server).

  Users on any cluster node can run print jobs, alter their priorities, and cancel them in the same way as they would if the spooler were running on their own system.
- Any printer or plotter can be "owned" exclusively by the cluster node it's attached to (when used as a raw device).

Examples later in this chapter illustrate adding a spooled printer to the cluster root server and to a cluster client.

Modems

Modems can be used only by someone logged in to the cluster node to which the modem is attached.

Modems for use with `cu(1)` or `uucp(1)` are supported only on the cluster root server and are not shared: you must be logged in to the server to use them.

Terminals

You can add terminals to cluster nodes and use them just as you would on a standalone machine.

- When adding a terminal to a cluster node (root server or client), you must create the device file and edit `/etc/inittab` while logged in to that node. This is because the `/dev` directory and `/etc/inittab` are context-dependent; logging in to the affected client means that you will, by default, edit the right version of the file. (See Chapter 2, "Understanding Clusters", in this manual, for an explanation of context-dependent files.)

  If you use SAM to add the terminal, SAM will create the device file and modify `/etc/inittab` for you, but, again, you must be logged in to the node to which you're adding the terminal.

  For instructions on adding a terminal, see the *Installing Peripherals* manual.

  For more information on `/etc/inittab`, see `inittab(4)` in the *HP-UX Reference*.

- Once you grant a login name and password to a terminal user, they are valid on every node in the cluster, not just on the node to which the terminal happens to be attached. (See Chapter 13, "Managing Users in a Cluster").

Adding Peripherals to the Cluster Root Server

This section contains an outline procedure for adding peripherals to the root server, followed by an example.

You can use any peripheral on a cluster root server that is supported by that model of computer. 12

In a cluster, however, you should think carefully about how you want to distribute peripherals between the root server and the clients. The preceding sections of this chapter explain the options and restrictions, and there's further discussion in the next section, "Adding Peripherals to a Cluster Client".

You add peripherals to the root server in much the same way as you would to a standalone computer, but you must keep two important points in mind:

- When configuring HP-UX to communicate with a peripheral attached to the server, you must be logged in to the server.
- When you reboot the server, you automatically reboot the entire cluster.

The major tool for adding peripherals is the **SAM** (System Administration Manager) utility.

The other manuals you will need are:

- *Installing Peripherals*
- *System Administration Tasks*

Note	This chapter deals only with peripherals such as disk drives, tape drives, printers and terminals. It does not explain how to configure the cards these peripherals attach to. For information on configuring cards, consult the *Installing Peripherals* manual. For E/ISA cards in particular, see Appendix A, "E/ISA Configuration", in that manual.

Summary of Steps

This is intended to give you an idea of the scope of the task, and of the differences between adding a peripheral on a cluster server and adding a peripheral on a standalone computer. If you are an experienced system administrator, you may find this summary sufficient.

If you are not experienced, or not used to adding peripherals, you should use SAM if possible, read the documentation as directed at each step, and read through the example that follows the summary.

1. Turn off the peripheral.
2. If you are going to attach the peripheral to an E/ISA card, make sure the card is configured.

   See Appendix A, "E/ISA Configuration", in *Installing Peripherals*, for directions.
3. If this is a SCSI device, shut down and turn off the cluster server.

   This will shut down the entire cluster.
4. Connect the peripheral to the root server.

   Set configuration switches on the peripheral if necessary, following directions in the manual that came with the peripheral.
5. Connect the peripheral to the power supply and turn the peripheral on.
6. Log in to the root server as superuser.
7. Configure HP-UX to communicate with the peripheral.
   - Create a device file for the peripheral.
   - If necessary, add the device driver for the peripheral to the kernel.
   - Do other configuration tasks that may be required for a given peripheral.

   The following will give you an idea of what's involved in each case.

Device File If you use SAM to add the peripheral, SAM will create the device file for you.

If you decide not to use SAM, follow directions for creating a device file in the *Installing Peripherals* manual. An example later in this chapter (under "Adding Peripherals to a Cluster Client") shows how to create a device file for a tape drive using the `mknod` command.

Device Driver If you use SAM, SAM will check whether the appropriate device driver is in the kernel, and add it if necessary.

If you decide not to use SAM, you'll find alternative procedures for adding the device driver in Chapter 14, "Setting Up Devices Using HP-UX Commands", in the *Installing Peripherals* manual.

Other Tasks Other configuration tasks include editing `/etc/inittab` for a terminal (see *Installing Peripherals*) or adding a printer to the line-printer spooler (see the *System Administration Tasks* manual, Chapter 9, "Managing Printers and Printer Output").

If you use SAM to add the peripheral, SAM will normally do these tasks for you.

8. If you (or SAM) added a device driver to the kernel, do the following:
   a. Rebuild the kernel.
   b. Install the new kernel.
   c. Reboot the cluster server.

   You can do these kernel tasks either via SAM or with HP-UX commands. The example below uses the SAM method. You'll find procedures for both methods in the *System Administration Tasks* manual, Chapter 2, "Constructing an HP-UX System".

   Chapter 11, "Reconfiguring the Kernel for a Cluster Node", in the present manual, summarizes special considerations for configuring a cluster server's or client's kernel.

Caution Rebooting the cluster root server will reboot all the clients as well. See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", in this manual, for details.

Example: Adding a Spooled Printer to the Cluster Root Server

Let's assume you have a Model 425S cluster server and you want to attach an HP *LaserJet III* printer to it. The *LaserJet* will be spooled as part of the **printer class** "laser" (that is, it will be one of a pool of printers with the group name "laser"). Its own printer name will be "laser3".

It will not be the **system default** printer, since another printer is already serving this function. In a cluster, the system default printer is the printer to which all print jobs, from all the computers in the cluster, are sent if the user does not specify a printer name.

In this example we'll connect the printer to the computer's parallel port, and we'll have SAM create the device file `/dev/ptr33459A`.

Before You Start

If you're unfamiliar with the line printer spooling system, read "Managing Printers and Printer Output", in the the *System Administration Tasks* manual.

Procedure

The following example shows how to add this printer using the SAM program.

1. Connect the printer to the cluster server.
   a. Make sure that the printer's power switch is set to OFF.
   b. Set configuration switches on the printer.

      (Consult the documentation that came with the printer for details.)
   c. Connect the printer cable to the printer and to the computer.
   d. Connect the printer's power cord to the printer and to the power supply.
   e. Turn on the printer.

2. Gather the information you'll need to supply to SAM.

   The following table shows the information needed and the values you would supply in this particular case. For explanations of the items, see "Managing Printers and Printer Output", in the *System Administration Tasks* manual, and the Help screens in SAM.

Information Needed	**Example**
Printer name	laser3
Printer model/interface	hp33447a
Printer device file name	/dev/ptr33459A
Printer priority ("Default Request Priority")	0
Make this the system default printer?	n
Printer class	laser

3. Run SAM *on the root server.*
4. Select:

   `Printers and Plotters ->`

   then

   `Printers/Plotters`

   Then choose

   `Add Local Printer/Plotter`

   from the `Actions` menu. and select the parallel interface.

Note SAM distinguishes between "local" and "remote" printers. In a cluster. what SAM calls a "local" printer is a printer that is physically connected to the computer on which you're running SAM: the cluster root server in this example. SAM uses the term "remote printer" to refer to a printer that is not connected to any computer in the cluster.

Note If the device driver is not configured into the kernel, SAM will ask you if you want to add it; you should accept.

Adding the driver involves regenerating and re-installing the kernel and rebooting the system. SAM will do these tasks for you if you say so, but remember that rebooting the root server will reboot the entire cluster.

12

5. Enter the values from the checklist into SAM.

   (If SAM has rebooted the server for you after installing a new kernel, you will need to get back into SAM and get to the `Add Local Printer/Plotter` menu.)

You have now added the printer to the root server and to the cluster's line printer spooling system. Any user on the server or any client can send print jobs to this printer.

You can also add a spooled printer to a cluster client, and that printer too will be accessible to any user on any cluster node. There's an example in the next section, "Adding Peripherals to a Cluster Client".

Adding Peripherals to a Cluster Client

This section explains how to add a peripheral to a cluster client.

Peripherals in a cluster may need to be used differently depending on whether they're attached to the cluster server or a client. Disk drives in particular raise many issues, which are dealt with under "Local Disks", later in this chapter. Tape drives and printers are more straightforward, but there are some points to keep in mind; these are summarized under "Tape Drives" and "Printers and Plotters" below.

If you have not already done so, you should also study the detailed rules and guidelines in the first section of this chapter, "How Clusters Support Different Peripherals".

Note

This chapter deals only with peripherals such as disk drives, tape drives, printers and terminals. It does not explain how to configure the cards these peripherals attach to. For information on configuring cards, consult the *Installing Peripherals* manual. For E/ISA cards in particular, see Appendix A, "E/ISA Configuration", in that manual.

Summary of Steps

The following summary is intended to give you an idea of the scope of the task, and of the differences between adding a peripheral on a cluster client and adding a peripheral on a standalone computer. If you are an experienced system administrator, you may find this summary sufficient. If you are not experienced, or not used to adding peripherals, you should use SAM if possible, read the documentation as directed at each step, and read through the examples later in this section.

1. Turn off the peripheral.
2. If this is a SCSI device, shut down and turn off the cluster client.

Caution

- If this is an auxiliary swap server, shutting it down will shut down all the swap clients.
- If this is an auxiliary file server, shutting it down will unmount any locally mounted file systems.

See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

---

3. If you are going to attach the peripheral to an E/ISA card, make sure the card is configured.

   See Appendix A, "E/ISA Configuration", in *Installing Peripherals*, for directions.

Caution

If you are connecting a disk drive to an E/ISA card, and the disk is to be used for local swap, you *must* configure the card before the client attempts to swap to the disk. If the card is not configured, the client will panic.

---

4. Connect the peripheral to the cluster client.

   Set configuration switches on the peripheral if necessary, following directions in the manual that came with the peripheral.
5. Connect the peripheral to the power supply and turn the peripheral on.

6. Boot the client if necessary and log in *to the cluster client* as superuser.

   Make sure you are logged in to the cluster client to which you are adding the peripheral.

7. Configure HP-UX to communicate with the peripheral.

   - Create a device file for the peripheral.
   - If necessary, add the device driver for the peripheral to the kernel.
   - Do other configuration tasks that may be required for a given peripheral.

   The following will give you an idea of what's involved in each case.

   Device File If you use SAM to add the peripheral, SAM will create the device file for you.

   If you do not use SAM, follow directions for creating a device file in the *Installing Peripherals* manual.

   (The example of adding a tape drive to a cluster client, later in this section, illustrates creating a device file with the `mknod` command on a Series 300 client.)

   Device Driver If you use SAM, SAM will check whether the appropriate device driver is in the kernel, and add it if necessary.

   If you decide not to use SAM, you'll find alternative procedures for adding the device driver in Chapter 14, "Setting Up Devices Using HP-UX Commands", in the *Installing Peripherals* manual.

   Other Tasks Other configuration tasks include editing `/etc/inittab` for a terminal (see the *Installing Peripherals* manual), or adding a printer to the line-printer spooler (see the *System Administration Tasks* manual, "Managing Printers and Printer Output").

   If you use SAM to add the peripheral, SAM will normally do these tasks for you.

8. If you (or SAM) added a device driver to the kernel, do the following:
   a. Rebuild the kernel.
   b. Install the new kernel.
   c. Reboot the cluster client.

   You can do these kernel tasks either via SAM or with HP-UX commands. The example under "Printers and Plotters", later in this section, uses SAM.

   You'll find procedures for both methods in the *Installing Peripherals* manual.

12

Caution Read the "Rules and Guidelines" in Chapter 11, "Reconfiguring the Kernel for a Cluster Node", in the current manual, before you reconfigure a cluster client's kernel.

Auxiliary Servers

- If this client is acting as an **auxiliary swap server** (if it has a local disk to which other clients are swapping), rebooting it will reboot the other clients that are swapping to this disk as well.

  See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", in this manual, for details.

- If this client is acting as an **auxiliary file server** (if it has a local disk that is being used for a locally mounted file system), shutting down the client will unmount the file system. See Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", in this manual, for details.

Tape Drives

The example that follows illustrates adding a tape drive to a Series 300 client. The following are important points to bear in mind.

- You add a tape drive to a cluster client in the same way as you would add it to the cluster server (assuming the same model of tape drive and computer in both cases).

  - To add a tape drive to Series 300 or 400 computer, look up the tape drive in the *Installing Peripherals* manual for the Series 300/400 and follow directions there and in the installation manual that came with the tape drive.

  The example in this section illustrates adding a 9144A cartridge drive to a model 370 cluster client.

Note Remember that you must perform all configuration steps (such as creating a device file, and adding a device driver to the kernel if necessary) while logged in to the client to which you are attaching the tape drive.

- Tape drives are not shared.

  To use a tape drive, you must be logged in to the computer to which the tape drive is attached.

  Before you decide whether to attach the tape drive to a client or to the server, consider the implications for software installation, update and backup. These are explained in the notes about tape drives under "How Clusters Support Different Peripherals", earlier in this chapter.

Example: Adding a Tape Drive to a Series 300 Cluster Client

In this example, we'll connect a 9144A cartridge tape drive to the built-in HP-IB interface of a Model 370 cluster client. (We'll assume the interface card is at select code 7, which is the default.) We will use HP-UX commands rather than SAM to configure the drive, but you can use SAM if you prefer (choose `Tape Drives` from the `Peripheral Devices` menu).

1. Make sure the driver `cs80` is configured into the cluster client's kernel.

   Log in to the client to which you are going to attach the tape drive, and enter:

   ```
   more /etc/conf/dfile
   ```

Note `/etc/conf/dfile` will reflect the running kernel of the client you are logged in to unless you have reconfigured the kernel from some other source file, and did not copy this source file to `/etc/conf/dfile`. HP recommends that `/etc/conf/dfile` should always match the running kernel; see Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for details.

You should see the `cs80` driver listed near the top of the file. This driver is part of the normal configuration for a Series 300 cluster client, but if for some reason it is not there, or if it's listed but with an asterisk in front of it, you will need to add it and rebuild the cluster client's kernel.

Full directions for reconfiguring a Series 300/400 kernel are in Chapter 14, "Setting Up Devices Using HP-UX Commands", in the Series 300/400 version of the *Installing Peripherals* manual, and specific guidelines for configuring a cluster client's kernel are in Chapter 11, "Reconfiguring the Kernel for a Cluster Node", in the current manual.

Caution

- Read the "Rules and Guidelines" in Chapter 11, "Reconfiguring the Kernel for a Cluster Node", before you reconfigure a client's kernel.
- Remember you must be logged in to the cluster client to modify its kernel.

---

2. Connect the cartridge tape drive to the cluster client.

   You'll find directions for connecting this tape drive in the manual that came with the drive, and configuration information in Chapter 7, "Installing Disk and Tape Drives", in the Series 300/400 version of *Installing Peripherals*. In this example, we'll do the following:

   a. Turn off power to the tape drive.

   b. Set the HP-IB bus address on the tape drive.

      In this example we'll use bus address 2.

      You set the address by setting the four switches labelled "ADDRESS" on the back of the drive. The switches are labelled 1 through 4 and the two positions for each are labelled 0 and 1.

      You set address 2 by setting switch number 3 to the "1" position and the remaining switches to the "0" position.

   c. Record the HP-IB address on the worksheet near the end of the *Installing Peripherals* manual.

   d. Connect the cable that came with the tape drive to the computer and to the tape drive.

   e. Connect the electrical power cord to the tape drive and your power source.

3. (Skip this step if there is already a device file for this drive at this address on this client.)

   Using information in the *Installing Peripherals* manual, create a device file for the tape drive.

   a. Make sure you are logged in to the Series 300 cluster client to which the tape drive is attached.

   b. Look up this tape drive model in Chapter 7, "Installing Disk and Tape Drives", in the Series 300/400 version of the *Installing Peripherals* manual.

   c. Use the information from the previous step to construct a `mknod(1M)` command.

      The command

      ```
      /etc/mknod /dev/rct/1s0 c 4 0x070200
      ```

      creates a character device file for a 9144A drive connected to the built-in HP-IB interface (which is at select code 7) and set to HP-IB address 2.

      Remember that this example may not reflect the way you have connected your drive. Use Chapter 7, "Installing Disk and Tape Drives", in the Series 300/400 version of the *Installing Peripherals* manual to see what information you need to pass to `mknod` in your particular case.

      To find out more about the `mknod` command, look up `mknod` in section 1M of the *HP-UX Reference*.

4. Turn on power to the tape drive.

The tape drive is now configured and ready for use. Remember that you must be logged in to this client to use the tape drive attached to it.

Printers and Plotters

The example that follows illustrates adding a spooled printer to a Series 300 cluster client. The example uses SAM (System Administration Manager) screens to add the printer, but you can also use HP-UX commands; see Chapter 9, "Managing Printers and Printer Output", in the *System Administration Tasks* manual. If you have not used SAM before, you'll find a tutorial in Chapter 1, "Introduction to System Administration", in the *System Administration Tasks* manual.

The following are points to bear in mind when adding a printer or plotter to a cluster client.

- The terms "local printer" and "remote printer", as the SAM program uses them, have the following meanings in a cluster.
  - A "local printer" is a printer attached to the particular computer we're talking about, whether this computer is the cluster root server or a client.

    For the sake of clarity, we'll say "client printer" when referring specifically to a printer attached to a cluster client.
  - A "remote printer" is a printer attached to a computer that is not a member of the cluster.
- A printer attached to a cluster client can be spooled, even though the spooler runs only on the cluster root server.
  - If you use SAM to add the printer, as in the example that follows, it will automatically be spooled.
  - If you don't want the printer to be spooled, you can still add the printer using SAM as in the example below, and then remove it from the line-printer spooler by means of the `lpadmin` command. See `lpadmin(1M)` in the *HP-UX Reference*

    You can also add an unspooled printer using HP-UX commands. Use the information and instructions in the *Installing Peripherals* manual. An unspooled printer is accessible exclusively to the user of the client to which it's attached. It cannot be shared by other members of the cluster.
- Spooled printers attached to clients behave the same as spooled printers on the server, though a little magic goes on behind the scenes to make it all work (see the explanation following the example).

Example: Adding a Spooled Printer to a Series 300 Cluster Client

Let's assume you have a Model 375 cluster client and you want to attach an HP *LaserJet* 2000 printer to it. The *LaserJet* will be spooled as part of the **printer class** "laser" (that is, it will be one of a pool of printers with the group name "laser"). Its own printer name will be "laser4". It will not be the cluster's **system default** printer, since a printer attached to the cluster server is already serving this function.

12

(The system default printer in a cluster is the printer to which all print jobs, from all nodes in the cluster, are sent if the user does not specify a printer name.)

In this example we'll connect the printer to the built-in nine-pin RS-232 port.

Assume this cluster client has the cluster **node name** `client1` (this is the name you entered into SAM when you added the client to the cluster.)

The following is a summary of the steps you need to take to add this printer.

Before You Start.

1. If you're unfamiliar with the line printer spooling system, read Chapter 9, "Managing Printers and Printer Output", in the *System Administration Tasks* manual.
2. Make a backup copy of the cluster client's kernel.

   If the driver for this printer is not in the kernel, SAM will add it for you, but SAM does not automatically make a backup copy of a client's kernel, so you should make one yourself:

   a. Log in as the root user.

   b. Copy the kernel file (`/hp-ux`) to a name that will identify this particular client, for example:

   ```
   cp /hp-ux /sysbckup.client1
   ```

Caution DO NOT copy the client's kernel to `/SYSBCKUP`. `/SYSBCKUP` contains the cluster server's backup kernel. See Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for more information.

Procedure.

1. Connect the printer to the cluster client.
   a. Turn off power to the printer.
   b. Set configuration switches on the printer
   c. Connect the printer interface cable to the printer.
   d. Connect the interface cable to the nine-pin RS-232 port.
   e. Connect the printer's electrical power cable to the printer and the power source.
   f. Turn on the printer.

   (Consult the manual that came with the printer for details.)

2. Gather the information SAM will need to add the printer to the line-printer spooling system.

   The following table shows the information needed and the values you would supply in this particular case. For explanations of the items, see Chapter 9, "Managing Printers and Printer Output", in the Series 300/400 version of the *System Administration Tasks* manual, and the Help screens in SAM.

   Given our example, the items and their values are:

Information Needed	Example
Printer name	laser4
Printer model/interface	hp2684a
Printer device file name	/dev/lp_laser4
Printer priority ("Default Request Priority")	0
Make this the system default printer?	n
Printer class	laser

3. Log in *to the cluster client* as superuser.
4. Run SAM *on the client to which you connected the printer.*
5. Select:

   `Printers and Plotters ->`

   then:

   `Printers/Plotters`

   Then choose:

   `Add Local Printer/Plotter`

   from the `Actions` menu, and:

   `Add Serial (RS-232) Printer/Plotter`

   from the pop-up menu.

6. SAM now checks to see if the device driver for this printer is configured into this client's kernel.

   - If the device driver is in the kernel, continue with the next step.
   - If the device driver is not configured into the kernel, SAM will ask you if you want to add it.

     You should accept, but first back up the cluster client's current kernel if you have not already done so. (See "Before You Start" just before this procedure. You'll need to get to a shell to do this now: if you're running SAM in a window, switch to another window and log in *to this cluster client* as root; if you're running SAM on a terminal, press the (Shell) function key.)

Note Adding the driver involves regenerating and re-installing the kernel and rebooting the system. SAM will do these tasks for you if you say so, but if this client has a local disk, remember:

- □ If other clients are swapping to the disk, those clients will be rebooted too.
- □ If the local disk is being used for a file system, you will not be able to reboot the client until all users and programs have stopped using that file system. Then shutting down the client will unmount the file system, and bringing the client back up will remount it.

---

7. Enter the values from the checklist into SAM.

   (If SAM has rebooted the client for you after installing a new kernel, you will need to get back into SAM and get to to the `Add Local Printer/Plotter` menu.)

   If you enter the name of a device file that does not already exist, SAM will create the device file for you.

8. Exit SAM.

What Happens Behind the Scenes. When SAM adds a spooled printer to a cluster client, as in the example above, it creates a **context-dependent file** `/usr/spool/lp/interface/`*printername*, where *printername* is the name you supplied to SAM.

(A context-dependent file is a file whose contents differ depending on which member of the cluster is using it. See Chapter 2, "Understanding Clusters", for more information.)

In the above example, SAM would create the context-dependent file `/usr/spool/lp/interface/laser4`, with two elements, `default` and `client1` (the node name we gave to the client that has the printer). The full path names for these two elements are `/usr/spool/lp/interface/laser4+/default` and`/usr/spool/lp/interface/laser4+/client1`.

SAM also creates another file, in this case `/usr/spool/lp/member/laser4`. This file contains the device file name you supplied on the SAM screen, `/dev/lp_laser4`.

When a user sends a job to printer `laser4` from any member of the cluster (the server or any cluster node, including `client1`), the following things will happen:

1. The `lp` command, as always, invokes the line printer scheduler (`lpsched`).

Note In a cluster, `lpsched` runs only on the server, whether or not there are client printers.

---

2. `lpsched` in turn executes the script contained in the `default` element of `/user/spool/interface/laser4`.
3. The script does the following:
   a. Figures out which client has the printer and waits until that client is ready to accept `remsh` (remote shell) commands.
   b. Gets the device file name for the printer from `/usr/spool/lp/member/laser4`.
   c. Builds a command which is passed via `remsh` to `client1`. This command executes the `client1` element of `/usr/spool/lp/interface/laser4` on `client1`.

      This `client1` element contains the printer model/interface script whose name you supplied to SAM: `hp2684a` in this case.
4. The `hp2684a` script sends the output file to the printer, just as it would on a standalone machine.

Local Disks

You can attach a disk to any client in the cluster. Such a disk is usually called a "local disk". Local disks are optional: you can attach all the cluster's disks to the server if you like, but there are cases where one or more local disks may be more efficient. Advantages and disadvantages of local disks are discussed below. 12

A local disk, like any other disk under HP-UX, can be used for file space, swap space, or both. There are restrictions, however. The most important of these are:

- File space is always shared.

  A client cannot have its own "private" file system.
- The root file system must always reside on the root server's disk.

The rules for local disks are spelled out in full near the beginning of this chapter. If you have not already read them, you should do so before you add a disk to a cluster client.

Locally Mounted vs. Centrally Mounted File System

Every member of a cluster shares the cluster's entire file system. A user on any node can get to files on any disk in the cluster, no matter which node (server or any client) the disk is attached to.

However, files stored on the server's (or on other clients') disks must be fetched over the LAN when a client needs them, and this can affect performance. If LAN traffic is causing unacceptable delays on your system, it's worth considering redistributing the disks.

Advantages and Disadvantages of a Locally Mounted File System.

Advantages:

- Efficiency.

  Can reduce LAN traffic, and speed application throughput.

- Flexibility.

  File-system disk drives can be attached to any cluster node.

Disadvantages:

- More complex to administer.
  - You need to understand the rules and restrictions governing locally mounted file systems. See "How Clusters Support Each Class of Peripheral", earlier in this chapter.
  - A client with a locally mounted file system must be shut down carefully, since shutdown will unmount the file system. See "Shutting Down an Auxiliary File Server" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

Where To Go from Here.

- The next section, "Local vs. Shared Swap", explains the options each client has for swap.
- If you decide to add local disks to one or more clients, for swap space, file space, or both, turn to the section "Example: Adding a Local Disk", later in this chapter.

Local vs. Shared Swap

When a cluster client swaps to a remote disk, it must send and receive large packets across the LAN. If a client is very busy (for example running one or more large or data-intensive applications), the overhead of remote swapping may make response time unacceptably slow. In this case, you may want to configure the client to swap to its own local disk.

In fact HP-UX clusters offer three options for client swapping. A cluster client can do any one of the following:

- Swap to the cluster server's disk space.
- Swap to another client's disk space (**distributed swap**).
- Swap to its own local disk space.

Of these options, the first—swapping to the server's disk space—is the easiest to understand and administer; the second gives you flexibility in distributing your resources; the third should provide the best performance for a given client.

(Remember that the cluster server must always swap to its own disk space.)

Advantages and Disadvantages of Local Swap.

Advantages:

- Efficiency.

  When a client swaps to its own disk, it does not incur the LAN overhead of swapping to the cluster root server's (or to another client's) disk space.

- Flexibility.

  Disk drives can be attached to any cluster node, and cluster clients can be configured to swap to any node's disk space (their own, the cluster root server's, or another client's).

Disadvantages:

- More complex to administer.

  A client whose swap space is being used by other clients must be shut down carefully, since shutdown will bring down those other clients also. See "Shutting Down an Auxiliary Swap Server" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

- Performance improvement is not guaranteed.

  A client with a slow local disk may have no better response time than a client swapping to a fast disk on the server; in fact, response time could be worse.

Where To Go from Here.

- If you want to explore the option of mounting a file system on a disk attached to a client, read the previous section, "Locally Mounted vs. Centrally Mounted File System."
- If you decide to add local disks to one or more clients, continue with the next section, which shows how to add a local disk and configure swap and/or a file system on it.
- If you want to enable swapping from a client to another client's disk space, and the target disk is already installed and configured, turn to the section "Setting Up Swap to an Auxiliary Swap Server", later in this chapter.

Example: Adding a Local Disk

In this example we'll connect a 7958 disk drive to the built-in HP-IB interface of a Model 370 cluster client. We'll assume that the device driver, `cs80`, is already configured into the Model 370's kernel (it is by default), but you may need to regenerate the kernel and reboot for other reasons.

12

Will You Need To Reboot?

SAM will prompt you to regenerate and install the client's kernel and reboot after adding the disk if:

- The disk will be used for swap and the client does not already have a disk that it is using for swap;

  or

- The disk will be used for a file system and the client is not already configured as an **auxiliary file server** (that is, if it does not already have a file-system disk that other members of the cluster are using).

SAM will do these tasks for you if you say so.

Before You Start.

- Read the rules and explanations under "Disk Drives" in the first section of this chapter, "How Clusters Support Different Peripherals".
- This procedure will require you to regenerate the kernel and reboot the client in many cases.

1. If you have not already done so, read Chapter 11, "Reconfiguring the Kernel for a Cluster Node", to get an understanding of what happens when you reconfigure a client's kernel.
2. Make a backup copy of the client's kernel.

Caution DO NOT copy the client's kernel to `/SYSBCKUP`. `/SYSBCKUP` contains the cluster server's backup kernel. See Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for more information.

Log in *to the cluster client* as superuser and copy the kernel to a name that will uniquely identify this client, for example:

```
cp /hp-ux /sysbckup.client1
```

Caution

If you are setting up local swap to a disk connected to an E/ISA card, you *must* configure the card before the client attempts to swap to the disk. Otherwise the client will panic.

- To configure an EISA card, do the following:
  1. Shut down and turn off power to the client.
  2. Connect the card.
  3. Turn power on and reboot the client.

  The system will configure the card automatically, and you can now connect the disk and enable swapping to it, as shown in the following example.

- To configure an ISA card, do the following:
  1. Shut down and turn off power to the client.
  2. Connect the card.
  3. Turn power on and reboot the client.
  4. Run `eisa_config` to configure the card, as described in Appendix A, "E/ISA Configuration", in *Installing Peripherals*.

  You can now connect the disk and enable swapping to it, as shown in the following example.

Procedure.

1. Shut down and turn off power to the client.
2. Turn off power to the disk drive.
3. Set the HP-IB bus address on the drive. 12

   You must set an address that is not already in use on the card (the built-in interface in this example).

   If you have been keeping the worksheet at the end of the *Installing Peripherals* manual up to date, you will be able to find the next available address from there. Otherwise you can make a physical check of the addresses set on the devices connected to this interface.

   You set the bus address by turning the wheel labelled "ADDRESS" on the back of the disk drive. In this example, we'll set it to 3.

Note When adding a disk drive to cluster client, avoid using bus address 0 (zero) if possible; if there is a boot area on the disk and the address is zero, the client will try to boot from the disk rather than over the LAN.

---

4. Record the address on the worksheet at the end of the *Installing Peripherals* manual.
5. Connect the HP-IB cable to the disk drive and to the cluster client computer.
6. Connect the electrical power cord and turn on power to the disk drive.
7. Turn on power to the cluster client computer and let it reboot.

8. Use SAM to configure the disk drive into the cluster.

   SAM is the menu-driven System Administration Manager supplied with HP-UX. If you haven't used SAM before, you'll find a tutorial in Chapter 1, "Introduction to System Administration", in the *System Administration Tasks* manual.

   a. Run SAM *on the cluster client to which you have just attached the disk drive.*
   b. Select:

      `Peripheral Devices ->`

      then:

      `Disks and File System(s) ->`

      then:

      `CD-ROM, Floppy, and Hard Disks`

   c. SAM will show the model number, select code and bus address of the disk you have just selected.

d. Choose

    Add a Hard Disk

from the Actions menu and supply the following information to SAM:

i. Whether you want to use the disk for file space, swap space or both. 12

ii. When to mount a file system and/or enable swapping to the disk.

- Turn off Now and and leave At Every System Boot turned on if you're going to reboot the client as a result of adding this disk. See "Will You Need to Reboot?" at the beginning of this example.
- If you won't need to reboot, leave both options (Now and At every System Boot) turned on.

iii. The mount directory (that is, the point in the cluster's file system "tree" at which you want to attach the file system on this disk, if any).

If the directory doesn't exist, SAM will create it for you.

iv. Whether you want SAM to create a new file system for you on the disk.

Caution Creating a new file system will destroy any files already on the disk.

If you ask SAM to create a file system, and you have chosen to use the disk for swap as well, SAM will prompt you to allocate the amount of swap space and file space.

When configuring swap space, as a rule of thumb, allow 30 megabytes for each client that will swap to this disk. There's more information on configuring swap in the *System Administration Tasks* manual, in Chapter 7, "Managing Swap Space".

Note When adding a disk drive to a cluster client, do not turn on the option to copy the bootstrap program onto the disk. If you turn it on, the client will try to boot from the disk. You want the client to boot from the LAN, not from the disk.

At this point, if you said the disk was to be used only for swap and this client is not already configured as an **auxiliary swap sever** or **auxiliary file server**, SAM asks you if you want to allow other nodes to use the disk.

If you say no, you will be able to use the disk *only* for local swap (that is, this client and no other will be able use the disk), and it will be more difficult to convert this local swap disk to a shared disk if you decide to do so in the future.

On the other hand, if you say here that the disk can be shared, there is nothing to prevent you from using the disk exclusively for this client's local swap.

Say you want other clients to be able to use this disk unless:

- You intend, now and in the future, to use the disk only for local swap

  and

- You are very short of core memory.

  If you say the disk can be shared, SAM raises the number of General Cluster Server Processes (GCSPs) running on the client. These take up a small amount of memory. See "What SAM Has Done", a little later in this chapter, for other changes.

e. If you need to regenerate the kernel and reboot, SAM will offer to do these tasks for you.

   Let SAM go ahead and do them, but first back up the cluster client's current kernel if you have not already done so. (See "Before You Start" just before this procedure. You'll need to get to a shell to do this now: if you're running SAM in a window, switch to another window and log in *to this cluster client* as root; if you're running SAM on a terminal, press the `Shell` function key.)

What SAM Has Done. When you use SAM to add a local disk, SAM does the following:

1. Creates a device file for the disk drive.
2. Calls /etc/newfs to create a new file system if you have so indicated. 12

Note

When configuring a disk for a Series 300 or 400 cluster client, SAM calls /etc/newfs with the -n option so as to create a file system without a boot area. If you opt to use HP-UX commands rather than SAM to build the file system on a Series 300/400 client's local disk, make sure that you use this -n option to newfs, or set the disk's HP-IB address to something other than zero; otherwise the client will try to boot from the boot area on the disk rather than over the LAN.

3. Calls /etc/mount to mount the file system, if any, at the mount point you indicate, if you chose Now for when to mount the file system.
4. Changes defaults for the new file system (such as permissions, long or short file names, etc.) if you have so indicated.
5. Adds an entry to /etc/checklist for the new file system and/or swap area (if necessary) if you chose At every system boot for when to mount the file system or enable swap. (/etc/checklist is a context-dependent file with an element for each cluster client).
6. Modifies this client's entry in /etc/clusterconf:
   - To indicate local swap, if you said you wanted to use the disk for swap and the client is not already swapping locally.
   - To set the number of **GCSPs** (**General Cluster Server Processes**) to 4 if you said other clients could use the disk, and if the number is currently less than 4. (A larger number will not be changed.)

     Cluster Server Processes are explained in Chapter 2, "Understanding Clusters".

   See Chapter 8, "Introduction to Cluster Administration", for an explanation of the entries in /etc/clusterconf.

7. Modifies the client's kernel:
   - To indicate local swap, if you said the disk was to be used for swap and the client is not already swapping locally.
   - To change kernel parameters if you said other clients could use the disk:
     - Sets `dskless_node` to 0.
     - Sets `num_cnodes` to 5.
     - Sets `ngcsp` to `8*num_cnodes` (40).
     - Sets `server_node` to 1.

8. Installs the new kernel.

   - If SAM regenerates and installs the kernel, the new kernel is in this client's portion (**element**) of `/hp-ux`.

     (See Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for an explanation of the structure of the context-dependent file `/hp-ux`.)

   - If SAM regenerates the kernel, but you opt not to have SAM install it, the new kernel is saved as `/etc/conf/hp-ux`, which is not a context-dependent file. See **Caution** below.

9. Saves the new kernel source file.

   - By default, SAM saves the new kernel source file in `/etc/conf/dfile`.

     `/etc/conf/dfile` is a context-dependent file.

   - If you don't allow SAM to overwrite `/etc/conf/dfile`, SAM saves the kernel source file as `/etc/conf/dfile.SAM`.

     `/etc/conf/dfile.SAM` is *not* a context-dependent file, and will be overwritten next time you use SAM to regenerate any cluster node's kernel.

Caution

- If you didn't opt to have SAM install the new kernel, you should install it yourself (or copy the kernel file `/etc/conf/hp-ux` to a unique name) before changing any other cluster node's kernel; `/etc/conf/hp-ux` is not a context-dependent file and will be overwritten next time you regenerate *any* cluster node's kernel, whether or not you use SAM.

- If you chose not to let SAM overwrite `/etc/conf/dfile`, you should move `/etc/conf/dfile.SAM` (which SAM saves instead) to `/etc/conf/dfile` as soon as possible.

See "Rules and Guidelines", in Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for a full explanation.

Where To Go from Here. The disk is now connected and configured.

If you have configured a swap area on the disk, and you want other clients to swap to it as well, continue with "Setting Up Swap to an Auxiliary Swap Server" below.

Things To Remember.

- If you have configured a file system on the disk:
  - You must be logged in to the client to which the disk is attached in order to run file-system maintenance commands such as `fsck`. These commands are listed under the rules for locally mounted file systems, near the beginning of this chapter.
  - You will not be able to shut down the client so long as anyone (or any program) has files open in the file system, because shutting down the client will unmount the file system.

    See "Shutting Down an Auxiliary File Server" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".
- If other clients will swap to this disk:
  - Shutting down this client will also shut down the other clients that are swapping to it.

    See "Shutting Down an Auxiliary Swap Server" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

Setting Up Swap to an Auxiliary Swap Server

This section explains what to do after you have set up swap on a cluster client's local disk, as described in the previous section, and now want to make other clients swap to it as well. A client to whose disk space other clients are swapping is called an **auxiliary swap server**, and we refer to this kind of swapping as **distributed swap**.

12

Remember these rules:

- An auxiliary swap server must swap to its own disk space (this can include swapping to a file system on a local disk).
- Only cluster clients can swap to an auxiliary swap server: the cluster root server must swap to its own disk space.

The rules for swapping in a cluster are spelled out in full under "Swap Space" in the first section of this chapter, "How Clusters Support Different Peripherals".

To enable swapping from a client or clients to an auxiliary swap server, do the following:

1. Configure swap on a disk (or disks) attached to the client that is to be the auxiliary swap server.

   The previous section explains how.

Caution A client that is currently swapping locally (but does not have any swap clients) may need to be reconfigured before other clients can swap to its disk. See "Converting a Client to an Auxiliary Swap Server" later in this chapter.

---

2. Log in as superuser on *the cluster root server* and run SAM.
3. Select:

   `Cluster Configuration ->`

4. For each client which is to swap to the auxiliary swap server:
   a. Highlight the client's name
   b. Select `Modify Swap Server...` from the `Actions` menu
   c. Highlight the name of the swap server from the list on the `Modify Swap Server` screen.
   d. Press (OK).

   For example, if you have configured swap on a disk attached to `client2` and you want `client1` to swap to this disk, highlight `client1` on the `Cluster Configuration` screen, select `Modify Swap Server` and then highlight `client2` on the `Modify Swap Server` screen.

   (This will change `client1`'s swap entry in `/etc/clusterconf`—see Chapter 8, "Introduction to Cluster Administration", for details of `/etc/clusterconf`.)

5. Halt and cycle power on each of the clients whose swap server has changed.

Caution **To avoid client panics:**

- Make sure the auxiliary swap server (the cluster node to which a given client will now swap) is up and running before you reboot the swap client.
- If you have changed the swap server for a client that was formerly swapping to another client's disk space, do not reboot the former swap server, and do not reboot the cluster, before you reboot the swap client. See "Rebooting to Change Swap Server" below.

Enter

```
/etc/reboot -h
```

then wait for the "halted" message and turn power off and on again. The client will now boot from the new swap server.

Note A simple reboot is not enough. You must either cycle power (turn the client off and on) or reboot with the `-l` option and specify the link level address of the new swap server's LAN card. for example:

```
/etc/reboot -l 0x08000900a63f
```

Rebooting to Change Swap Servers. Each cluster node's swap server is recorded in `/etc/clusterconf` (see Chapter 8, "Introduction to Cluster Administration"). Changing a client's swap server changes its swap entry in `/etc/clusterconf`.

When you shut down (or halt or reboot) a cluster client, the system checks `/etc/clusterconf` to see if this client has any swap clients. If there are swap clients, they are shut down gracefully (see Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes"); if not, just this cluster client is shut down.

For example, if `client1` and `client2` in Figure 12-1 (near the beginning of this chapter) are each swapping to their own local disk, or to `server`'s disk, then shutting down `client1` will have no effect on `client2`, and *vice versa*.

But if `client1` is swapping to `client2`'s disk space, then shutting down `client2` will shut down `client1` as well.

A problem can arise when you have just changed a client's swap server and you reboot the former swap server.

For example, if `client1` has been swapping to `client2`'s disk, and you have just changed `client1`'s swap server to make it swap to the root server's disk instead, *DO NOT* shut down, halt or reboot `client2` without first shutting down `client1`. If you shut down `client2` first, `client1` will panic with an error such as:

```
devswap_pagein: syncpageio detected an error
```

This comes about as follows:

1. You change client1's swap server in SAM, from client2 to server.
2. SAM changes client1's entry in /etc/clusterconf to show that client1 will now swap to the root server.
3. You issue the shutdown command on client2.
4. The system checks /etc/clusterconf to see if client2 has any swap clients.
5. /etc/clusterconf shows no swap clients for client2.
6. The shutdown completes on client2.
7. client1 attempts swap to client2's disk.

   client1 has not been rebooted and so it continues to attempt to swap to client2.
8. The swap fails because client2 has been shut down.
9. client1 panics.

The above scenario could occur even if you do a cluster-wide shutdown (by shutting down the root server), because the root server may shut down client2 before it gets to client1.

To be sure of avoiding swap-related panics, always proceed as follows:

1. Use SAM to change the client's swap server.
2. Make sure the new swap server is up and running.
3. Reboot and cycle power on the client whose swap server has changed.
4. Reboot any other cluster nodes you need to reboot.

For example (given the sample cluster shown in Figure 12-1, near the beginning of this chapter), if client1 has been swapping to client2's local disk space, and you have just changed the configuration to make both clients swap to server's disk, you should:

1. Halt client1 and cycle power. Client1 will come up and start swapping to server's disk space.
2. Do the same on client2, which will also now swap to server's disk space.

Converting a Client to an Auxiliary Swap Server

The best way to configure an auxiliary swap server is to add the local disk in SAM, as described earlier in this chapter under "Example: Adding a Local Disk".

But if the client is already using the disk for local swap, you can't use SAM to change the disk's use from exclusive (used only by this client) to shared (used by other clients). You could remove the disk (after first reconfiguring the client to swap elsewhere) and then use SAM to add the disk back, but you probably don't want to do that.

Follow the directions in this section if all of the following are true:

- The client is currently swapping to its own local disk.
- You want to allow other clients to swap to the disk.
- When you added the disk, you did not say other nodes could use the disk (see "Example: Adding a Local Disk", earlier in this chapter).

  This could mean either that you explicitly rejected this option, or that you never saw it (you might have added the disk before it was possible to share a client's disk, or you might not have used SAM at all).

What You Are Going To Do. You are going to edit the cluster configuration file, `/etc/clusterconf`, to raise the number General Cluster Server Processes (GCSPs) running on this client. Then you'll use SAM to change kernel parameters, reconfigure and re-install the kernel, and reboot the client.

What You Need Before You Start.

- Read the "Rules and Guidelines" for reconfiguring a cluster node's kernel, in Chapter 11, "Reconfiguring the Kernel for a Cluster Node", in this manual, and also read through the example, later in that chapter, that shows how to change kernel parameters. Put a marker there if you're not familiar with SAM.
- Find the detailed description of the layout of `/etc/clusterconf`, in Chapter 8, "Introduction to Cluster Administration", under "Cluster-Specific Files". Put a marker there as well.

First, Edit /etc/clusterconf.

1. Log in on any cluster node.
2. Edit `/etc/clusterconf` in your favorite editor, for example

```
vi /etc/clusterconf
```

3. Find this client's entry in the file. The entry will contain the client's **node name** (the name by which this client is uniquely known in the cluster).

   For example, if the client's node name is `client1`, the line will look something like this:

```
080009000565:2:client1:c:2:1
```

4. Change the number of GCSPs.

   This is the number right at the end of the line. In the example just given, the number is 1, and it will probably be 1 in your case. Change it to 4.

   Given our example, the line will now read:

```
080009000565:2:client1:c:2:4
```

5. Save the file.

Now Reconfigure the Kernel.

1. Log in as superuser *on the client to which the disk is attached.*
2. Copy `/hp-ux` to a backup file (*not* `/SYSBCKUP`). For example:

   `cp /hp-ux /sysbckup.client1` *(Use a name that will identify this client)*

Caution `/SYSBCKUP` is reserved for the cluster server's backup kernel.

---

3. Run SAM and get to the `Configurable Parameters` screen (under `Kernel Configuration`).

12

4. Make the following changes:
   - Change `dskless_node` to 0 (zero).
   - Change `ngcsp` to 40.

     (This is the maximum number of GCSPs that can run on this client, whereas the value you entered into `/etc/clusterconf` is the default number that will be started whenever you reboot the client.)
   - Change `num_cnodes` to 5.
   - Change `server_node` to 1.
5. Before you can exit SAM, SAM will warn you that the kernel has been changed and offer to regenerate and re-install the kernel.

   Let SAM regenerate and install the kernel and reboot the system for you.

Caution If you don't opt to have SAM install the kernel, you must install the new kernel yourself (or save the kernel files under unique names) before changing the kernel of any other cluster node.

See Chapter 11, "Reconfiguring the Kernel for a Cluster Node", for instructions.

What To Do Next. Once the client is up and running, you can use the SAM to make other clients swap to this disk, as described earlier in this chapter, under "Setting Up Swap to an Auxiliary Swap Server".

Remember that once other clients are swapping to this disk, shutting down this client will also shut down the other clients that are swapping to it.

See "Shutting Down an Auxiliary Swap Server" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes".

13

Managing Users In A Cluster

Managing users in a cluster is much like managing users on a single, multi-user system, but some of the implications of the situation may not be immediately apparent. This chapter spells out these implications.

Users and Files in a Cluster

The most important thing to remember about users in a cluster is that *each login name is valid for the entire cluster.*

This means:

- A user who can log in on one cluster node can log in on all cluster nodes.
- A user who can access a file or directory from one cluster node can access it from all.

Note As far as user-management goes, think of the cluster as *one system*: manage users of a cluster as you would users of a standalone, multi-user system.

Example: A User's Capabilities in a Cluster

If a person named Fred has a workstation on his desk which is part of an HP-UX cluster, and you set up a user `fred` for him with a home directory `/users/fred`, then Fred will be able to:

- Log in to his own workstation as `fred`.
- Use the files in `/users/fred`.

  These files can be on any disk configured into the cluster.
- Use any other files, programs and commands whose permissions do not exclude him.

  It doesn't matter where those files reside physically: all files on all devices mounted to the cluster are available to him (and any other user of the cluster), subject to permissions.

  In a cluster, there is no such thing as a "local file". File systems may be *mounted* locally (from a client's local disk) or globally (from the server's disks), but they are *available* globally, to all users of the cluster who have the appropriate permissions.
- Use peripherals attached to his workstation.
- Use peripherals, such as printers and plotters, which are not attached to his workstation but have been configured to be globally available to users of the cluster.

  There's detailed information about peripherals in Chapter 12, "Adding Peripherals to a Cluster".
- Log in as `fred` (remotely or directly) to any other workstation in the cluster, including the cluster server, and use the same files, programs and commands as he can from his own workstation.

  (In the case of **context-dependent files**, he will now see the elements that pertain to this other cluster node. See Chapter 2, "Understanding Clusters" for a full explanation of context-dependent files.)

  He can now also use any devices (such as tape drives) which are exclusively available to this other cluster node.

13

Superuser

The superuser login, like all other login names, is valid on all computers in the cluster.

You, the system administrator, should have superuser capabilities, and you will probably want to nominate a deputy to cover for you in your absence. This person will need the superuser password.

No other user of the cluster needs to know, or should know, the superuser password. Users of the workstations in the cluster do not need superuser capabilities to run their computers.

Because your superuser login name is valid cluster-wide, you can log in as superuser on any cluster client as the need arises.

Allowing Users Shutdown Capabilities

By default, only the superuser can shut down a system with the `shutdown(1M)` command. You may decide that you want to extend this capability to other cluster users, to allow them to shut down their own workstations. You can do this by means of the `/etc/shutdown.allow` file. Instructions are in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes", under "Shutting Down a Cluster Client".

Special Privileges

As on a standalone system, you can assign special privileges that are usually reserved to the superuser, to other groups of users if you need to.

The privileges in question are RTPRIO (which allows one to set real-time priorities); MLOCK (which allows one to lock processes in memory); LOCKRDONLY (which allows one to lock a file that is open for read-only); SETRUGID (which allows one to change the real user or group ID of a process); and CHOWN (which allows one to change ownership of files).

You assign these privileges by means of the `setprivgrp (1M)` command; see the *HP-UX Reference* for details. 13

In a cluster, these group privileges apply *only to the cluster node on which you set them*. For example, if you want group `patrick` to have the RTPRIO privilege on cluster clients `client1` and `client2`, you must execute the `setprivgrp` command twice: on `client1` and again on `client2`.

Adding and Removing Users

You add and remove users in a cluster in the same way as you would on a standalone system. You can be on any cluster node when you add or remove a user and the result will be the same.

Chapter 4, "Controlling Access to the System", in the *System Administration Tasks* manual, provides full directions for adding and removing users.

File Security

If your users are moving from standalone workstations to a cluster, you will probably need to help them readjust their thinking about security. Their files are no longer isolated: every other user of the cluster can see them.

To recreate the level of security afforded by a standalone workstation, every user would need to deny everyone else read, write and execute permissions on their files.

Fred could do this quite easily for the files in his home directory by entering:

```
chmod 700 /users/fred
```

However, one of your reasons for setting up a cluster may have been to allow people to share files more easily, so you probably don't want security to be this strict.

The model to keep in mind is a standalone, multi-user system. For security purposes, the cluster as a whole is like one minicomputer or mainframe and the individual workstations are like terminals. You and your users should protect all files just as you would on a multi-user system.

Context-Dependent Files and Security

Context-dependent files (CDFs) can *by default* be seen only from a cluster node with a matching context, but our hypothetical user Fred can use an explicit reference to get to any context-dependent file not protected by permissions.

For example, if Fred's workstation has the node name `client1`, and if the context-dependent file `/users/applic/prog1.src+` has only a single element `client2`, then by default Fred will not be able to see this file from his own workstation. But if he enters the command

13

```
more /users/applic/prog1.src+/client2
```

he will be able to read the file if it is not read-protected. Or he could log in to `client2` and enter

```
more /users/applic/prog1.src
```

with the same result.

From this you can see that you should never make a file into context-dependent file in order to keep its contents private: this is not what CDFs are for. Use standard HP-UX file permissions to protect files.

There's a full explanation of context-dependent files in Chapter 2, "Understanding Clusters".

14

Updating A Cluster

Use this chapter and the *Installing and Updating HP-UX* manual to do either of the following:

- Update the cluster from one release of HP-UX to another.
- Install or update application software in a cluster.

You must be on HP-UX release 8.0 or later to update to 9.0

Since "update" is a term that can have many meanings, check the table on the next page to make sure you're on the right track.

To accomplish this	Do this
Install HP-UX for the first time onto a computer intended to be an HP-UX cluster server	1. Install HP-UX with the help of *Installing and Updating HP-UX*. 2. Use Chapter 4, "Setting Up a Cluster", in this manual, to help you configure the computer as a server.
Convert a running system to a cluster server	1. Update the system to Release 9.0 with the help of *Installing and Updating HP-UX*. 2. Configure the system as a cluster server following directions in Chapter 4, "Setting Up a Cluster", in this manual.
Add a new computer to the cluster	Follow directions in Chapter 5, "Adding Cluster Clients", in this manual.
Update the cluster from one release of HP-UX to the next	Turn to the section "Updating HP-UX" in this chapter.
Add new application software to the cluster	Turn to the section "Installing and Updating Applications" in this chapter.
Update application software already on the system	Turn to the section "Installing and Updating Applications" in this chapter.

Updating HP-UX

By "updating HP-UX" we mean updating the operating system itself, for example from Release 8.0 to Release 9.0. To do this you use the `update(1M)` utility.

Operating system releases are bundled with HP-supplied subsystems such as networking, and you will normally update these subsystems when you update HP-UX, although the `update` utility allows you to choose exactly what you want to update. (Updating the networking software does not pose any special problems in a cluster.)

Updating HP-UX in a **homogeneous cluster** (Series 300/400 only) is very much like updating a standalone machine. Use the procedure that follows.

If You Have Local Disks

`Update` needs to have access to HP-UX system directories. If these directories are on a client's local disk, and you have shut down the cluster prior to running `update` as we recommend, the update will fail.

In general, do not move system directories from the root server's disk. If you think you or someone else may have moved some of them, but you're not sure which, the following checklist should help.

For the purposes of this rule, there are two classes of directory:

1. Directories which should reside in the root server's disk space, but whose children can reside on a client's local disk.

   The directories in question include:

   ```
   /bin
   /bin/posix
   /dev
   /dev/rdsk
   /dev/x29
   /etc
   /etc/frm_scr
   /etc/hhg
   /etc/update.lib
   /etc/x25
   /lib
   /lib/libp
   /tmp
   /usr
   /usr/doc
   /usr/news
   /usr/old
   /usr/pub
   ```

2. Directories which should reside in the root server's disk space, and whose children should also reside in the server's disk space.

   The directories in question include:

   ```
   /etc/conf
   /etc/interface.lib
   /etc/net
   /etc/newconfig
   /system
   /usr/add-on
   /usr/adm
   /usr/bin
   /usr/diag
   /usr/etc
   /usr/include
   /usr/lib
   /usr/man
   /usr/netdemo
   /usr/nettest
   /usr/sam
   /usr/spool
   /usr/src
   /usr/tsm
   ```

Procedure: Updating the Cluster

Follow these steps to update the cluster.

Do this on the cluster server.

1. If you're going to be updating from tape or CD-ROM (rather than from a `netdist` server), make sure that the cluster server has the right type of drive attached and configured.
2. Log in to the *cluster server* as superuser.

   To run `update(1M)`, you must be logged in to the cluster root server and you must be superuser.
3. Make sure that all other users of the cluster have saved their files and logged out.

   You do not need to halt or reboot any of the cluster clients.
4. Shut down the cluster so that you are in single-user mode on the cluster root server.

   Shutting down the root server shuts down the whole cluster. The clients will wait for the cluster server to reboot. (See "Shutting Down a Cluster" in Chapter 10, "Booting and Shutting Down Clusters and Cluster Nodes" if you need more detail.)
5. Update the HP-UX system on the cluster server following directions in Chapter 5, "Updating HP-UX", in *Installing and Updating HP-UX*.

   Remember that you must load the tapes or CD-ROM disk on a drive attached to the cluster server.
6. After `update` reboots the server, the clients will reboot automatically and your users can log back in.

Installing and Updating Applications

This section explains how to install and update application software in the cluster.

To install or update application software, first read the "Rules and Guidelines" below, then continue with the section labelled "Procedure".

Rules and Guidelines

- Some application application software can be installed and updated via the HP-UX `update(1M)` utility; and some cannot. Always follow the supplier's instructions.
- You must run `update(1M)` from the cluster root server.

  If the software supplier's directions call for some other installation method (`tar` or `tcio | cpio` for example), you don't have to be logged in to the cluster root server, but it's simpler if you are.
- If you are loading an application from a device such as a tape or CD-ROM drive (rather than over a network), the drive must be attached to the computer from which you are running the installation program (normally the cluster root server).
- HP-UX system directories must be available to `update`. See "If You Have Local Disks" earlier in this chapter.

Procedure: Installing an Application

To install or update an application in a cluster, do the following:

1. Read the "Rules and Guidelines" earlier in this chapter if you have not already done so.
2. Read the software supplier's directions for installing or updating the application.
3. Make sure you have the appropriate installation device (tape or CD-ROM drive, etc.) attached and configured on the cluster root server.
4. Log in to the cluster root server as superuser.

   (Required if you are using `update(1M)`; recommended otherwise.)
5. Follow the software supplier's directions for loading and customizing the application.

15

Applications in a Cluster

Read this chapter before creating or buying applications to run in an HP-UX cluster. You should also read the sections on installing applications in Chapter 14, "Updating a Cluster".

How Clusters Affect Applications

An HP-UX cluster is different from both a single-user workstation and a multi-user system. You need to be aware of these differences before you install applications in your cluster. This section contains a brief sketch of the kinds of differences a programmer must take account of, and provides a set of rules and guidelines for cluster applications.

Clusters vs. Workstations

From a programming point of view, the most important difference between an HP-UX cluster and a standalone workstation is that in a cluster a program cannot assume that only one user will run the application in the same file system at the same time. A program that uses no file locking, for example, may always run correctly on a standalone workstation, but could corrupt data files in a cluster if more than one user runs it at the same time and one set of updates is interleaved with another.

Clusters vs. Multi-User Computers

A cluster is more like a multi-user system than it is like a single-user workstation, but there are still some differences, the most important of which is the I/O mechanism.

Response Time

On a cluster client with no local disk, both file I/O and swapping are going across the LAN (and through layers of networking and cluster protocol as well). This may result in slower response time than the same application would provide on a multi-user system, though often it does not because cluster caching algorithms eliminate much of the I/O delay.

New Applications. If you are thinking of buying an application to use in a cluster, and response time is critical, insist on benchmarking the application in a cluster (preferably *your* cluster): don't rely on results derived from a standalone system.

When commissioning applications in-house, make sure the programmers read and understand the rules and guidelines in the next section.

Existing Applications. In many cases, of course, the problem is how to make an existing application run faster.

Whether end-users actually experience slow response depends on many factors, but in a cluster, as in any computer network, there is one simple rule of thumb: the busier the LAN, the more likely you are to hear complaints about slow response. Chapter 3, "Setting Up the Network", has guidelines for setting up the optimum network for a cluster.

If you need to improve performance, but can't isolate your cluster from a busy LAN, the answer may be to distribute local disks so that the cluster clients both swap and do their file I/O to their own disks. Read the sections on local disks in Chapter 12, "Adding Peripherals to a Cluster", if you are considering this option.

Rules and Guidelines for a "Cluster-Smart" Application

Use the following as criteria when choosing, or designing, applications to run in a cluster.

15

- Applications must use file-locking to prevent the same file being updated simultaneously by different users.
- Applications should not open a file for writing unless they are actually going to write to it.

  When a cluster node opens a file that another node has open for writing, file caching on the clients is automatically turned off. This helps to safeguard file integrity, but degrades performance, so it's important to ensure it only happens when necessary.
- Applications designed to take advantage of different kinds of floating-point hardware must be compiled separately and installed into context-dependent files or directories. See Chapter 8, "Introduction to Cluster Administration", under "Creating a CDF", for an example.
- Application design must take account of the following:
  - IPC (Inter-Process Communication) mechanisms such as semaphores, messages and shared memory cannot be distributed across a network.

This means if you want processes to communicate with each other *across cluster nodes*, you cannot use System V IPC as a vehicle. You can, however, use other methods such as sockets and named FIFOs.

- Process priorities are not distributed in a cluster.

  Priorities are limited to a single CPU, so you can't depend on them, for example, to control the flow of updates to a given file, if the processes acting on that file are running on different cluster nodes.

Index

D

G

N

O

P

R

U

W

X

HEWLETT® PACKARD

Reorder No. or
Manual Part No.
B1864-90015

Manufacturing
Part No.
B1864-90015

B1864-90015