Diskless Workstation Administration Guide

Document Number 007-0855-030

CONTRIBUTORS

Written by Pam Sogard
Edited by Sylvia Thompson
Production by Derrald Vogt
Illustrations by Cheri Brown
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson, Erik Lindholm, and Kay Maitz
Engineering contributions by Michael Nishimoto
Other contributions by Glenda Ramil, Paul Reilly, Kim Simmons, Yvonne Wilborn, and Joe Yetter

Diskless Workstation Administration Guide
Document Number 007-0855-030

Contents

Figures

Tables

Examples

Using This Guide

The *Diskless Workstation Administration Guide* explains how to configure and maintain Silicon Graphics® software so that one or more networked workstations can operate without a local disk. It introduces the terms and concepts needed to understand a diskless implementation and explains how to plan an implementation. It also describes the procedures needed to install, maintain, and troubleshoot a diskless implementation.

Product Requirements

Silicon Graphics’ implementation of Diskless Workstation software should be used with these requirements in mind:

- Diskless Workstation software must be used in conjunction with Silicon Graphics' Network File System (NFS™) software option.
- To implement 5.1 or later versions of IRIX on diskless clients, the diskless server must be running IRIX Release 5.1 or later.
- The Diskless tools that install IRIX 5.1 software cannot be used to install earlier versions of IRIX. See the *Release Notes* that accompanied this product for instructions on installing pre-IRIX 5.1 software.
- The server and client systems must be on the same physical and logical network: the diskless client-server relationship cannot be implemented by means of a router.

Summary of Contents

The *Diskless Workstation Administration Guide* is organized into seven chapters and an appendix:

Chapter 1, "Understanding Diskless Operation"

Introduces the elements of a diskless network and explains the organization of diskless software on the server. Read Chapter 1 to understand how diskless implementations work.

Chapter 2, "Preparing for Installation"

Explains how to get the hosts in a LAN ready for a diskless implementation and describes how to organize diskless software directories. Use Chapter 2 as a guide to setting up a diskless implementation for the first time.

Chapter 3, "Creating a Share Tree"

Explains how to configure and install the software that is shared by diskless clients. Use Chapter 3 when installing a new share tree. Do not use Chapter 3 to update an existing tree with a new software release.

Chapter 4, "Creating Client Trees"

Explains how to configure and install the software that is used by an individual diskless client. Use Chapter 4 when installing a new client and swap tree. Do not use Chapter 4 to update an existing tree with a new software release.

Chapter 5, "Updating Software on Diskless Trees"

Explains how to upgrade a diskless software installation to a new software release and how to add software options to a diskless implementation.

Chapter 6, "Maintaining Diskless Implementations"

Contains instructions for performing routine server maintenance. Use the information in Chapter 6 to move a diskless server or software to a new location, remove a server or diskless software from service, or reconfigure the client with local resources.

Chapter 7, "Troubleshooting Diskless Installations"
Describes problem-solving techniques for diskless setups and suggests how to get the best results from service calls to Silicon Graphics' Technical Assistance Center.

Appendix A, "Determining the Client Architecture"
Provides supplementary information for specifying workstation hardware in the client configuration file. Refer to Appendix A if the hardware information for a client workstation is not listed in either Chapter 2 or Chapter 4 of this guide, or in the distribution copy of the client.dat file.

The Audience for This Guide

The *Diskless Workstation Administration Guide* is written for system and network administrators who will be responsible for managing a Silicon Graphics server that supports one or more diskless workstation clients. It assumes that the audience has experience with basic system administration on an IRIX™ or other UNIX-based system.

Experience in maintaining IRIX filesystems and NFS, and in installing Silicon Graphics software are helpful, but not required, to install and maintain Diskless Workstation software. A familiarity with various families of Silicon Graphics workstations is also helpful, particularly for doing troubleshooting procedures. At some sites, familiarity with network administration will be required to prepare network software for the diskless implementation.

Notation Conventions

This document uses the standard UNIX® convention when referring to entries in IRIX documentation. The entry name is followed by a section number in parentheses. For example, *cc(1)* refers to the *cc* manual entry in Section 1 of the *IRIX User's Reference Manual, Volume 1.*

In-text descriptions, file and directory names, as well as commands, are shown in *italics*; and options appear in **bold**. Command syntax descriptions appear in `typewriter font`, with variable names in *italics*. User entries are shown in **`typewriter bold`**.

Related Documentation

The documents listed below provide supplementary information on topics covered in the *Diskless Workstation Administration Guide*.

IRIS Software Installation Guide
IRIX Advanced Site and Server Administration Guide
NFS Administration Guide
NIS Administration Guide
The *Owner's Guide* for your client and server systems

Product Support

Silicon Graphics, Inc., provides a comprehensive product support maintenance program for its products. Refer to the *Diskless Workstation Release Notes* for product support information.

Chapter 1

Understanding Diskless Operation

This chapter explains the fundamentals of diskless operation. It contains the following sections:

- "The Diskless Client-Server Model"
- "The Diskless Tree"
- "The Influence of Client Classes"

The Diskless Client-Server Model

Diskless software enables computers on a local area network (LAN) to operate using the disk space, operating system, and support software supplied by a remote host. The *client* (the computer using the resources) must be on the same physical network as the *server* (the computer supplying the resources*).* The client-server relationship is dependent upon network protocols to boot the client over the network link, and upon IRIX, IRIX utilities, and NFS services to sustain the client after it is booted.

Figure 1-1 illustrates the resources that a server and its clients contribute to diskless operation.

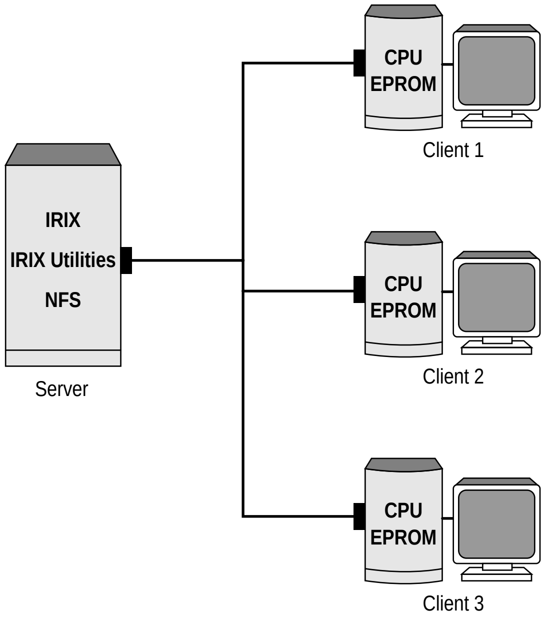

Figure 1-1 Requirements for Diskless Service

PROM Resources

The PROM on a diskless client must include the Internet Bootstrap Protocol, *bootp*(1M), the network protocol that initiates remote booting. During the client's power-up sequence, *bootp* sends a request for a bootfile (the *unix* image) to the diskless server. The server downloads the client's boot file using the Trivial File Transfer Protocol, *tftpd*(1M). (The details of this process are explained in "Debugging the Boot Process" in Chapter 7.)

IRIX and IRIX Utilities

Client installations of IRIX and its utilities are maintained in dedicated directories on the server. With the exception of the kernel, which resides in client memory after it is downloaded, the client retrieves and executes operating system programs and utilities from the server on an as-needed basis. Swap space needed for program execution is also furnished by the server (see “Local Disk Resources” on page 3 for exceptions).

NFS Resources

With the exception of the *unix* boot file, all files needed by a diskless client, including the IRIX operating system and IRIX utilities, are provided by NFS’s export and mount facilities. For this reason, a diskless server must be running NFS to export the necessary directories, and the NFS application must be available to each diskless client so that it can mount the exported directories. In addition to the required filesystems that it mounts from the server, a diskless client can also mount NFS filesystems and directories from other remote hosts.

Local Disk Resources

A diskless client can be equipped with a local disk of its own. The local disk on such a client does not supply the minimal operating system, and the disk is not used for booting, as it is on conventional standalone systems. Rather, the client obtains booting and operating system resources from the server, and the local disk is reserved for specialized uses. Commonly, the disk is configured to provide local swap space or the */tmp* directory for the client.

The Diskless Tree

The software on a server that is dedicated to supporting its diskless clients is referred to as the *diskless tree*. Optimally, the diskless tree occupies an entire filesystem, but a diskless tree can also be configured as a portion of an existing filesystem.

All diskless trees are composed of three basic elements: *share tree*s, *client trees*, and *swap trees*. The overall size and configuration of a diskless tree depends on the number of clients it supports and the number of IRIX versions it offers.

Figure 1-2 illustrates the diskless tree for a server supporting three diskless clients on a single version of IRIX.

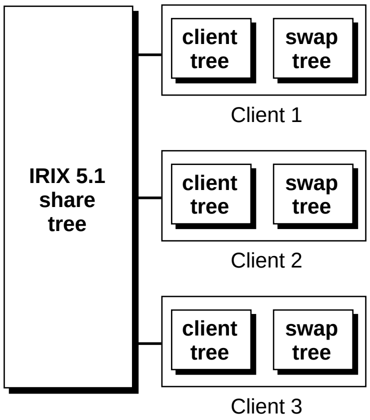

Figure 1-2 A Simple Diskless Tree

The Share Tree

The share tree contains the portion of the IRIX operating system that can be shared by a group of diskless clients. This includes IRIX kernel modules, hardware-specific files for each workstation model that the client supports, IRIX commands, and IRIX files that cannot be modified. The purpose of the share tree is to minimize the amount of disk space that the diskless implementation requires.

A diskless tree contains one share tree for each release of IRIX that it supports. The number of share trees that a server can support depends on the amount of disk space it contains and the usage patterns of its clients.

Figure 1-3 illustrates what happens to the diskless tree shown in Figure 1-2 when an additional IRIX release is added.

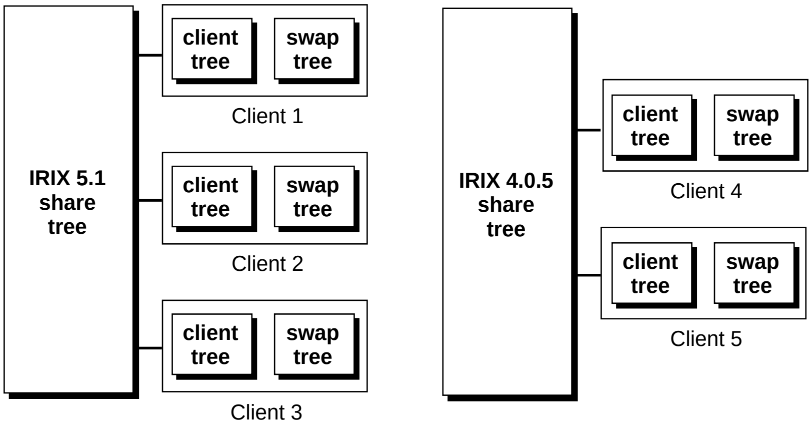

Figure 1-3 A Diskless Tree with Multiple Share Trees

The Client Tree

A client tree contains the IRIX kernel for an individual client, the specific hardware files that the client requires, and private files such as */etc/hosts*, */etc/fstab*, and */etc/sys_id*. All other operating system files reside on the share tree. The diskless tree contains a client tree for each client that it supports.

The Swap Tree

The swap tree provides the disk space that an individual client needs for virtual memory operations. It contains a single file whose contents support the virtual memory function of that client. The diskless tree typically contains a swap tree for every client that it supports; however, if a client contains a disk, the local disk may supply its swap space.

The Influence of Client Classes

A *class* is a group of workstations that contain identical CPU and graphics hardware. A class is generally (but not always) designated by a model name, such as the Indy or Indigo Elan.

A client class requires its own hardware-specific software modules. For this reason, a share tree must contain all the hardware-specific modules needed to support every class in its client base. By contrast, a client tree contains only those hardware-specific modules needed to support the individual client that is using it.

Chapter 2

Preparing for Installation

This chapter explains what to do to prepare for diskless software installation. It describes what to have on hand for the installation, what to check out and change to get ready for the diskless service, and how to organize the server filesystem that will contain diskless software. It also provides an overview of diskless software and the installation procedure.

This chapter contains the following sections:

- "Obtaining a Software Distribution"
- "Checking a Server"
- "Checking a Client"
- "Assigning Internet Addresses"
- "Planning the Diskless Tree"
- "Working with Diskless Software"
- "Overview of the Installation Procedure"

Obtaining a Software Distribution

The share trees, client trees, and swap trees should be created from an official release of Silicon Graphics software. Silicon Graphics software is originally distributed on compact disc read-only memory (CD-ROM). On some networks, a copy of this software is available from a central distribution point on the network. You can use either source to install diskless software, as long as you use an official software release.

At minimum, your IRIX 5.1 source must include the Silicon Graphics Execution Only Environment products, *eoe1* and *eoe2*; and the NFS product option, *nfs*. A group of subsystems from each of these products is required to install diskless software.

Checking a Server

Use the questions that follow to check the suitability of a server on which you plan to install the diskless tree. If you determine that any procedure described in this section is necessary, complete the procedure before you begin diskless installation.

1. Is the server reliable?

The ideal choice for a diskless server is a network host that is able to provide uninterrupted service to its clients for extended periods of time. Hosts that are subject to frequent shutdowns, such as laboratory systems, those used for software builds, or hosts subject to interruptions in service, are not likely to prove satisfactory as diskless servers.

2. Does the server have ample disk space?

On average, approximately 200 megabytes (Mb) of disk space is required for a diskless tree with these characteristics:

- One share tree of IRIX 5.1 with a minimum software configuration
- Two client trees with a minimal software configuration
- Two swap trees, each containing 20 Mb of swap space

Approximately 115 Mb is required for the same configuration, if the share tree contains IRIX 4.0. See “Planning Share Trees” on page 16 for more specific space planning information.

3. Is the server running the necessary software?

The IRIX version implemented on a server must be equal to or greater than the IRIX version implemented on any share tree that it supports. See the *Release Notes* shipped with your software for specific compatibility information.

To determine the version of IRIX running on a server, use the *uname*(1) command with the release option (*-r*), as shown in the following example. Do not install a share tree with a higher revision level than appears in the output from this command:

```
% uname -r
5.1
```

Table 2-1 shows the software subsystems that must be installed on a server to support clients using a share tree of IRIX 5.1 and IRIX 4.0.

Table 2-1 Software Requirements on Diskless Servers

Share tree	Product	Subsystems
IRIX 5.1	eoe1	eoe1.sw.lboot eoe1.sw.lib eoe1.sw.unix
	eoe2	eoe2.sw.tcp
	nfs[a]	nfs.sw.nfs nfs.sw.nis nfs.sw.dskless_server
IRIX 4.0	*eoe1*	All
	eoe2	Default selections
	nfs	nfs.sw.nfs nfs.sw.dskless

[a] If you are installing IRIX 4.X share and client trees, you must also install the *nfs.sw.dskless_server_405* subsystem. See the release notes for this product for installation instructions.

4. Is the TFTP environment set up correctly?

Client boot files are transferred from the server by means of TFTP. The server must contain a user account that owns TFTP (see *tftp*(1C)). By default, the owner of TFTP is *guest*. If you want a user other than *guest* to own TFTP, you must modify the file */etc/inetd.conf* (see *inetd*(1M)). If you plan to install software by downloading it from a remote host, be sure that TFTP security is correctly configured on the remote system (see *tftpd*(1M)).

5. Is an NIS domain name assigned?

To support diskless clients, a server must have an NIS domain name, whether or not it is running the network information service (NIS). To determine whether a server has an NIS domain name, use the *domainname*(1) command:

```
% domainname
my.domain.name
```

If this command gives no output, the server has no NIS domain name. It must be assigned a domain name (even a fictitious one) for diskless software to operate correctly. To set a domain name, issue the command shown in this example as the superuser.

```
# domainname fakedomain
```

If you would like the domain name to be set automatically when the server is rebooted, issue this command:

```
# domainname > /var/yp/ypdomain
```

6. Are there NFS export conflicts?

Be sure that the parent directory of your diskless tree is not listed as an NFS export. This condition causes NFS conflicts, which will either prevent the installation scripts from exporting diskless directories correctly or compromise security on the exported files. Check the server's */etc/exports* file to verify that the parent directory of the diskless tree is not listed for export.

7. Is the notation in the host database correct?

The host addresses in */etc/hosts* must be given in domain dot notation (see *hosts*(4)). For example, your entries in *hosts* should look something like this:

```
192.0.2.5 starlite.fakedomain.com starlite
```

8. Is the server on the same network as the client?

See the information about this question in “Checking a Client” on page 11.

Checking a Client

Use the questions below to check the suitability of a client for diskless operation. Some changes that might be required as a result of this checkout are changes you can make yourself; others might require additional members of your technical team. Any changes indicated by this check must be made before you begin diskless installation.

1. Is the client on the same network as the server?

Diskless service cannot be implemented by means of a router; the client and server systems must be on the same *physical network* (all hosts are connected to the same network cable). The physical network can be segmented by repeaters, however. If you do not have responsibility for cabling computers to the network, see your data communications technicians to determine whether a client and server are on the same physical network. If client and server are not on the same network, one system must be connected to the network of the other to implement diskless service.

The client and server must be members of the same *logical network* (the network portion of the internet address of the server and clients are the same) or subnetwork, if you are using subnetting. If you are not responsible for assigning internet addresses to hosts on the network, check with the site or network administrator who has this responsibility to verify that client and server network addresses are on the same network. If they are not, request a new address for either the client or server.

2. What hardware components does the client contain?

When you prepare the share tree configuration file, it helps to know model names of your diskless workstations and the designations for the hardware components that the workstations contain. You can use the *hinv* (1M) command at client workstations to get component information. Make a note of these components if you need to.

Table 2-2 shows the components of Silicon Graphics workstation models.

Table 2-2 Workstation Hardware Components

Model Name	CPU Processor	CPU Board	Graphics Subsystem	Graphics Subgroup
4D/20 and 4D/25	R3000	IP6	ECLIPSE	ECLIPSE
4D/30	R3000	IP12	ECLIPSE	ECLIPSE
4D/30 Elan	R3000	IP12	EXPRESS	EXPRESS
4D/35	R3000	IP12	ECLIPSE	ECLIPSE
4D/35 Elan	R3000	IP12	EXPRESS	EXPRESS
Indigo and Indigo Dual Head	R3000	IP12	LIGHT	LIGHT
Indigo XS and Indigo Elan	R3000	IP12	EXPRESS	EXPRESS
Indigo R4K and Indigo R4K DH	R4000	IP20	LIGHT	LG1MC
Indigo R4K XS, Indigo R4K, and Indigo Elan	R4000	IP20	EXPRESS	EXPRESS
Indigo^2 Extreme and Indigo^2 XZ	R4000	IP22	EXPRESS	EXPRESS
Indigo^2 Extreme+XL	R4000	IP22	NEWPRESS	NEWPRESS
Indigo^2 XL	R4000	IP22	NEWPORT	NG1
Indy	R4000	IP22	NEWPORT	NG1

Note: Additional information about workstation architecture is provided in the */var/boot/client.dat* file, and in Chapter 4 and Appendix A of this guide.

3. Does the EPROM support diskless operation?

All IP12, IP17, IP19, IP20, and IP22 model workstations contain EPROMs that support the *diskless* variable, a toggle that must be set to enable diskless operation. However, some early workstation models contain EPROMs that do not support this variable.

Table 2-3 shows EPROM revision levels that support the *diskless* variable in early models of Silicon Graphics workstations.

Table 2-3 EPROM Revision Requirements

Processor Type	EPROM Required
IP5	Revision 007
IP6	Revision E
IP7	Revision 007
IP9	Revision 007

Checking EPROMs with the *version* Command

You can determine the version of an EPROM in a client workstation by entering the *version* command at the Command Monitor prompt, as described in the following procedure.

Note: Early versions of the EPROM do not support the *version* command. If you are checking a client that does not support *version*, use the procedure in “Checking EPROMs without the version Command” on page 14 instead of this procedure.

1. Do a system shutdown; then power up to the System Maintenance menu.
2. Select option 5, “Enter Command Monitor,” from the menu.
3. Enter the *version* command and check its output:

   ```
   >> version
   ```

   If you get output like either of the samples below, then the EPROM on this workstation supports diskless operation.

   ```
   PROM Monitor Version 3.2 Rev E (followed by date and time)
   PROM Monitor Version 4D-4.0A (followed by time and date)
   ```

   Note: If you see the following output, this workstation does not support the *diskless* variable:

   `Version 3.3 IP6` (followed by time and date)

Refer to the “EPROM Workaround” in Chapter 7 to see whether the workaround is a solution for this workstation.

Checking EPROMs without the *version* Command

If a client workstation does not support the *version* command, you can determine whether its EPROM supports diskless operation by setting the diskless variable and checking the success of your setting:

1. Do a system shutdown; then power up to the System Maintenance menu.
2. Select option 5, “Enter Command Monitor,” from the menu.
3. Set the *diskless* parameter to *1*:

   ```
   >> setenv diskless 1
   ```
4. Type **init**.

   After your entry, the screen flashes; then the System Maintenance menu reappears.
5. Select option 5, “Enter Command Monitor,” from the menu.

   Check the setting on the *diskless* parameter:

   ```
   >> printenv
   ```

   If the *diskless* parameter is displayed and it is set to *1*, the EPROM supports the *diskless* variable.

Note: If a workstation you plan to use as a client does not support the *diskless* variable, see “EPROM Workaround” in Chapter 7 for recommendations, or call your support provider to request an EPROM upgrade.

Assigning Internet Addresses

Every client in a diskless implementation must be assigned a unique internet address so that it can be recognized as a network host. On some networks, the addresses of hosts with which a server communicates (including its diskless clients) are maintained in the server’s */etc/hosts* file. On networks using NIS or DNS services, diskless server and client entries must be added to the database used by these service.

If you are not the administrator for these services, advise the appropriate person to make the necessary entries for your diskless hosts. If you plan to assign internet addresses yourself and need more information, see the instructions for adding hosts to a network in one of these Silicon Graphics publications:

- *IRIX Advanced Site and Server Administration Guide* — for networks using */etc/hosts* and DNS
- *NIS Administration Guide* — for networks using NIS

Planning the Diskless Tree

The diskless tree is implemented as server directories that contain a share tree, client trees, and swap trees for each release of IRIX that a server supports. When these directories are well organized, diskless software is easier to install and maintain than when directories are organized in a random fashion. The directory organization scheme in this section can serve as a guide to organizing your own diskless tree.

The Diskless Filesystem

If possible, a diskless tree should be contained in its own filesystem; for large installations, an entire disk might be dedicated to the diskless tree. Using a dedicated filesystem makes administering NFS software easier. If the diskless tree is not in its own filesystem, the parent directory of the diskless tree should not be listed in the */etc/exports* file (see “Checking a Server” on page 8 for details).

Figure 2-1 shows the organization and naming of the first- and second-level directories of a filesystem for the diskless tree.

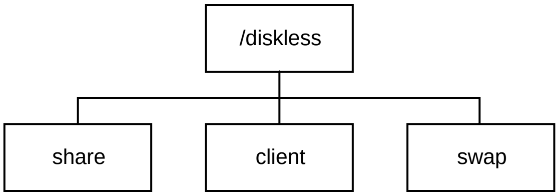

Figure 2-1 Level 1 and 2 Directories in the Diskless Trees

The top directory of the diskless tree, */diskless* in Figure 2-1, is referred to as the *root* of the diskless tree. Similarly, */diskless/share*, */diskless/client*, and */diskless/swap* are considered the root of the share, client, and swap trees, respectively.

Using Multiple Filesystems

It is possible to distribute elements of the diskless tree across multiple filesystems. To use more than one filesystem for a diskless tree, you simply adjust the variable settings in the *share.dat* file (see *DISKLESS* and *SHARE* descriptions in "Setting Share Tree Variables" in Chapter 3) and *client.dat* file (see *CLROOT* and *SWAP* descriptions in "Setting Client and Swap Tree Variables" in Chapter 4).

Planning Share Trees

The root directory for share trees, */diskless/share* in Figure 2-1, contains a share tree for each release of IRIX that the server supports. Each share tree contains the files normally located in the */usr*, */sbin*, */lib*, and */stand* directories on a standalone workstation.

Figure 2-2 illustrates the directory structure and typical contents of a share tree.

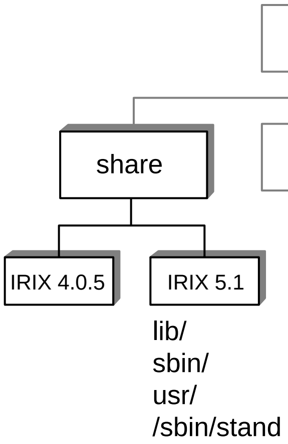

Figure 2-2 Typical Share Tree Contents

The subsystems that you install in a share tree greatly influence disk space requirements. To save space on the server, do not include applications in the share tree; plan to mount them as NFS directories from a server where they are already installed.

Table 2-4 provides an estimate of the disk space required to install a share tree for a class of Indigo^2 clients.

Table 2-4 Space Approximations for a Basic Share Tree

Software Product	Subsystems Installed	Space Needed
IRIX Execution Only Environments (*eoe1* and *eoe2*)	All default selections except manuals	90 Mb
NFS (*nfs*)	nfs.sw.nfs nfs.sw.nis nfs.sw.dskless_client	13 Mb
Default Window Manager (*4Dwm*)	All default selections except manuals	3 Mb
Desktop Environment (*desktop_eoe*)	All default selections except manuals	8 Mb
Motif™ (*motif_eoe*)	All default selections except manuals	3 Mb
X Window System™ (*x_eoe*)	All default selections except manuals	29 Mb
Total Requirement		146 Mb

The share tree in Figure 2-2 offers basic diskless functions and a windowing environment on the client workstations. If you need additional space-requirement information, see the space requirements listed in the release notes that accompany the software you plan to install.

Planning Client Trees

The root directory for client trees, */diskless/client* in Figure 2-1, contains a directory for each client that it supports. Each client tree contains the files normally located in the root (/) directory on a standalone workstation.

Figure 2-3 illustrates the organization of client tree directories and the typical contents of a client tree.

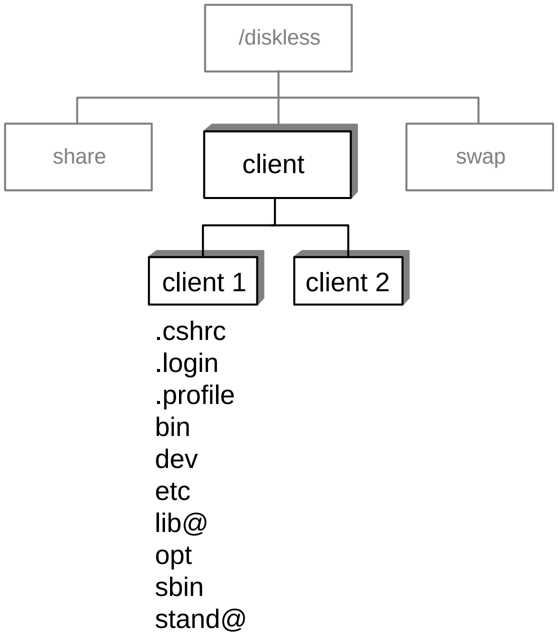

Figure 2-3 Typical Client Tree Organization and Contents

A newly installed client tree requires approximately 4 megabytes of disk space. However, since the client tree allows read/write access, space requirements may increase as a result of usage patterns on the client. This increase should be considered in your planning.

To save disk space and simplify software maintenance on the diskless server, mount */usr/people* and */var/tmp* from a different NFS server. This configuration makes it easier to upgrade system software and relocate user files when these changes become necessary.

Planning Swap Trees

The root directory for swap trees, */diskless/swap* in Figure 2-1, contains a swap tree for each client that it supports. The swap tree for an individual client is a directory containing a raw data file that provides swap space for virtual memory functions. By default, the size of the swap set up during diskless installation is 20 Mb. But swap space size is a configurable variable that you can set during the installation procedure and change after the swap tree is installed.

To determine the amount of swap space to allocate, consider the application requirements of your users. In cases where they do not require much disk space—when electronic mail and news readers are the principal applications, for example—you might consider reducing swap space to 10 megabytes. In cases where swap space usage is heavy—when users run compute-intensive applications, for example—consider increasing the default swap space size or adding a local swap disk to the client workstation (see Chapter 6 for details).

Figure 2-4 illustrates the organization of swap tree directories.

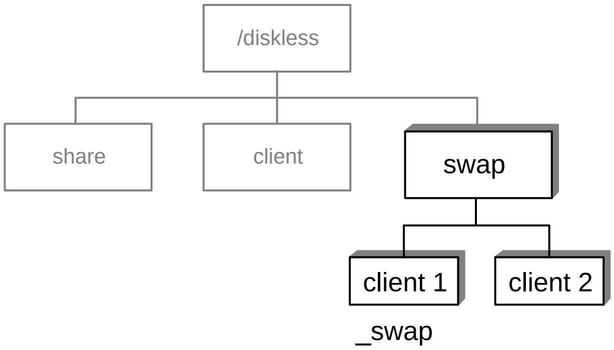

Figure 2-4 Typical Swap Tree Organization

Working with Diskless Software

Diskless software consists of two installation utilities and two configuration files. This software sets up the share tree, client trees, and swap trees for a diskless implementation:

/var/boot/share_inst

The program that creates a share tree using information in the share tree configuration file.

/var/boot/share.dat

The generic share tree configuration file that defines the location and contents of the share tree and specifies server variables that affect share tree installation. The server contains a separate working copy of *share.dat* for each version of IRIX that the server supports.

/var/boot/client_inst

The program that creates individual client trees and swap trees using information in the client tree configuration file, the share tree configuration file, and information that the installer provides when starting the program.

/var/boot/client.dat

The generic client tree configuration file that defines the location and contents of individual client trees. It also defines the location and size of the swap tree for each client. The server contains a working copy of *client.dat* for each group of client workstations that contain the same hardware components.

Using *inst* to Install a Diskless Tree

Both *share_inst* and *client_inst* invoke *inst* (the standard Silicon Graphics installation program) to install or upgrade the share and client trees. Once *inst* is invoked, you use it in the same way as you do to install software on standalone systems. When you install a client tree, *inst* automatically uses the software installed on the share tree as the software source.

Automatic File Updates

As part of the installation process, *share_inst* and *client_inst* automatically edit the server's */etc/exports* file, making the share tree, client trees, and swap trees available to diskless clients. These programs also automatically create a copy of the */etc/fstab* file in the client tree to add entries that mount the required directories from the server. Finally, *share_inst* and *client_inst* automatically edit the */etc//bootparams* file on the local host or the NIS master server if you are running NIS.

Overview of the Installation Procedure

The installation procedure is always done on the host that will act as the server system. The procedure consists of five basic steps:

1. Edit a working copy of *share.dat* to describe the server setup.
2. Run *share_inst* to install the share tree.
3. Edit a working copy of *client.dat* to describe the client tree for a class of workstations.
4. For each client, run *client_inst* to install its client and swap tree.

   Note: To save time on step 4, you can use a *client_inst* option called *cloning*, which builds client and swap trees without invoking *inst* (see Chapter 4 for details).
5. Boot each client and verify the installation.

If you have more than one client class, you need to repeat steps 3, 4, and 5 for each class that is being installed. If you are installing multiple IRIX releases, you need to repeat the entire procedure for each IRIX version that you install.

Figure 2-5 illustrates the flow of tasks in the installation process. Notice that the share tree is always installed before any of its client trees. This order is critical, since client trees are built from software contained in the share tree. The swap tree for a client is automatically built at the same time as its client tree.

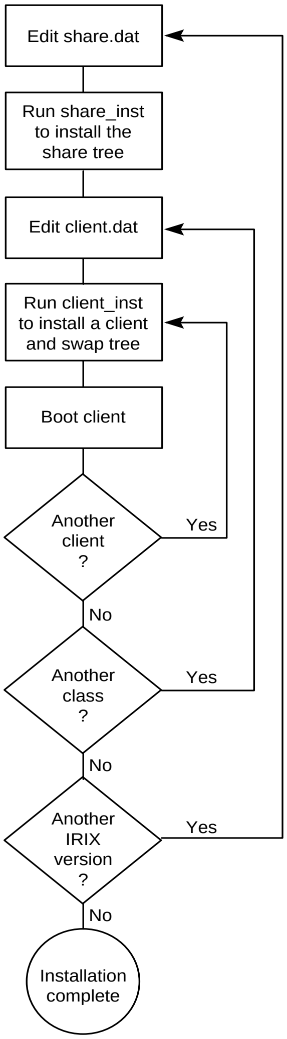

Figure 2-5 Flow Diagram of Installation Tasks

Chapter 3

Creating a Share Tree

To install software in a share tree, you create a working copy of the share tree configuration file, */var/boot/share.dat*, and edit the working copy to specify the requirements of your share tree. Then you run the share tree installation script, */var/boot/share_inst*, which installs share tree software using the specifications in your working copy of *share.dat*.

This chapter provides procedures for share tree installation. It contains the following sections:

- "The Share Tree Configuration File"
- "Creating a Working Copy of the Share Tree File"
- "Editing the Share Tree File"
- "Executing the Share Tree File"
- "Installing the Share Tree"

The Share Tree Configuration File

The configuration file for a share tree is a customized copy of the */var/boot/share.dat* file. The *share.dat* file defines the operating parameters that apply to all diskless clients using a particular share tree. The server must contain a separate working copy of *share.dat* for each share tree that it supports.

Creating a Working Copy of the Share Tree File

Your server should already have a distribution copy of *share.dat* in the */var/boot* directory as a result of installing Diskless software. Do not edit the distribution copy of *share.dat*; leave the distribution copy in its original form and make a working copy of the file to enter your changes. By convention, a working copy of *share.dat* is named for the release of IRIX that the share tree contains, and it ends with a *.dat* extension.

The following example illustrates how to create and name a working copy of *share.dat* for an IRIX 5.1 share tree.

```
# cp /var/boot/share.dat /var/boot/IRIX_51.dat
```

Editing the Share Tree File

In the as-shipped copy of *share.dat*, variables are assigned default values. Some default values, though not all, must be changed to reflect conditions in your diskless network. These variables provide three types of information: information about the diskless server, information about the share tree, and information describing the hardware components of client workstations.

The subsections that follow illustrate each section of *share.dat*. Below each illustration is an explanation of variables in the section and recommendations for setting them. Use these subsections as a guide when editing your working copy of *share.dat*.

Setting Server Variables

The server variables in *share.dat* define conditions on the server that affect diskless client operation. Example 3-1 illustrates the server specification section of *share.dat*.

Example 3-1 Server Specification Section of *share.dat*

```
# SERVER ENVIRONMENT SETUP
#
#       HOSTNAME   :  Do not change unless you understand network topologies.
#                     This variable should contain the name of the interface
#                     that is connected directly to the client machines.
#                     For example, a server machine “foobar” has two interfaces.
#                     Its primary interface is FDDI, and its secondary interface
#                     is ethernet.  The client machines are on an ethernet
#                     subnet connected to foobar’s ethernet interface.  foobar’s
#                     ethernet interface is named “ether-foobar”.  In this
#                     case, the HOSTNAME variable below needs to be set to
#                     “ether-foobar” because hostname will return “foobar”.
#       SHAREHOST  :  Do not change
#       SERVER_NIS :  yes - server is running NIS.Implies NIS lookup.
#                     no -- server is using local naming service.
#
HOSTNAME=”‘hostname‘”
SHAREHOST=”$HOSTNAME”
SERVER_NIS=”no”
```

The variables in the server specification section have these meanings:

HOSTNAME Specifies the host name of the diskless server. The default setting for *HOSTNAME* is the value returned by the *hostname* command. In most cases, the default setting is appropriate and should not be changed (see the note that follows for exceptions).

Note: If your diskless server has multiple network interfaces, *HOSTNAME* must be set to the name assigned to the network interface where the diskless LAN is attached.

For example, suppose your server has two network interfaces: an FDDI network interface with the host name *calvin*, and an Ethernet interface to the diskless LAN with the host name *ether-calvin*. In this example, the correct setting is *HOSTNAME=”ether-calvin”*.

SHAREHOST Specifies the hostname of the server. The default setting is the value specified by the *HOSTNAME* variable. Do not change this value.

SERVER_NIS Specifies whether to obtain the client’s host name and address from the NIS database. The default setting is *no*.

Set *SERVER_NIS* to *yes* if you want *share_inst* to obtain client addresses from the NIS hosts database. When you set *SERVER_NIS* to *yes*, *share_inst* makes an entry for this client in the bootparams map on the NIS master server if the master server is a Silicon Graphics system (see **Hint** that follows).

If you set *SERVER_NIS* to *yes*, client entries are added to the bootparams map on the NIS master server when the share tree is installed; but, you should expect some delay before the client entry is propagated to other NIS servers.

Set *SERVER_NIS* to *no* if you want *share_inst* to obtain client addresses from the server's local */etc/hosts* file. If you set *SERVER_NIS* to no, *share_inst* will make an entry for this client in the local server’s *bootparams* file.

Hint: If the NIS master is not a Silicon Graphics system and you want a client entry in the master’s NIS bootparams map, set *SERVER_NIS* to *no*; then, after you run *share_inst*, move the *bootparams* lines from the diskless server’s */etc/bootparams* file to the map on the NIS master.

Note: Diskless Workstation software installation scripts do not perform name and address resolution by means of DNS. However, DNS is supported after the share and client trees are installed.

Setting Share Tree Variables

Share tree variables in *share.dat* identify the server directory that contains the share tree for a given release of IRIX. Example 3-2 illustrates the share tree specification section of *share.dat*.

Example 3-2 Share Tree Specification Section of *share.dat*

```
# SHARE TREE SETUP
#
#       DISKLESS  : directory where all the share and client trees reside.
#       SHARE     : Do not change
#
DISKLESS=”/diskless”
SHARE=”$DISKLESS/share/$RELEASE”
#
```

The variables in the share tree specification section of *share.dat* have these meanings:

DISKLESS Specifies the pathname to the root of the diskless tree; that is, this directory contains the subdirectories for the share tree, client trees, and swap trees (see Figure 2-2).

Set *DISKLESS* to the pathname of the root of your diskless tree.

Note: When you execute *share_inst*, it reads only the first 37 characters of the combined characters in *SHARE* and *DISKLESS*, so you might need to limit the character-count of this variable.

SHARE Specifies the pathname to the root of the share tree (see Figure 2-2). The default setting for *SHARE*, $DISKLESS/*share*/*$RELEASE*, specifies the root of the share tree. Do not change this setting.

The default setting assigns $*DISKLESS* the value that you specified for *DISKLESS*, the previous variable. This setting assigns *$RELEASE* the name of the share tree, which you supply as an argument to the *share_inst* command when installing the share tree (see“The share_inst Command” on page 34).

Setting Client Hardware Variables

The client hardware variables specify the architectural components of the workstation models in the diskless network. These variables control the hardware-specific files that are included in the share tree.

A share tree must contain all the hardware-specific modules needed by the client classes represented in your client base. During client tree installation, the hardware specific modules that an individual workstation needs are selected from the share tree and installed into its client tree.

Figure 3-1 illustrates the hardware-specific modules that a share tree must contain to build the client trees for four classes.

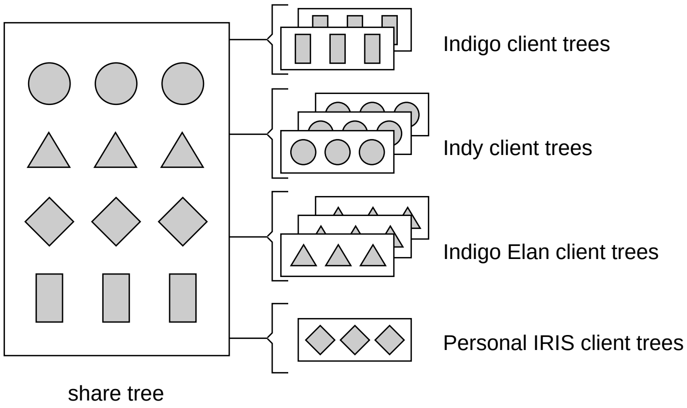

Figure 3-1 Hardware Modules for Multiple Classes

By default, *share_inst* populates the share tree with all the hardware-specific files for every Silicon Graphics workstation model. However, by setting the variables in this section of your working *share.dat* file, you can limit the hardware-specific files in a share tree to those files that your client models require.

Example 3-3 illustrates the hardware specification section of *share.dat*.

Example 3-3 Hardware Specification Section of *share.dat*

```
# CLIENT ARCHITECTURE SETUP
#
# The following variables are set using the table provided in client.dat.
# Please consult that file for valid values.
# The following 5 variables should not be set if all architectures are
# to be installed for the share tree.  This is why they are
# commented out.  If only a subset is to be installed, all 4 must be
# set.  The example below is for an R4000 Indigo with LIGHT graphics.
#
#CPUBOARD=”IP20”
#CPUARCH=R4000
#MACH=”IP20”
#GFXBOARD=”LIGHT -mSUBGR=LG1MC”
#VIDEO=” ”
#
# To install multiple architectures, set the 5 variables as shown below.
# Notice that the initial symbols CPUBOARD, CPUARCH, and GFXBOARD are
# imbedded in the values for the respective variables.  MACH only needs
# the “-m” flag imbedded.  The example below contains an R4k Indigo with
# LIGHT graphics and an R3k Indigo with XS/XS24/Elan graphics.
#
#CPUBOARD=”IP20 -mCPUBOARD=IP12”
#CPUARCH=”R4000 -mCPUARCH=R3000”
#MACH=”IP20 -mIP12”
#GFXBOARD=”LIGHT -mSUBGR=LG1MC -mGFXBOARD=EXPRESS -mSUBGR=EXPRESS”
#VIDEO=” ”
```

Note: Use Table 2-2, Example 4-2, or the tables in the *client.dat* file to determine the components in your client workstations. Be sure you include the components of all workstation models that will serve as diskless clients.

The variables in the hardware specification section of *share.dat* have these meanings:

CPUBOARD Specifies the CPU subsystems in client workstations. You can assign these values to *CPUBOARD*:

```
IP6
IP12
IP20
IP22
```

Set *CPUBOARD* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings. Check the samples in Example 3-3 (or the *client.dat* file) for the correct format of your settings.

*CPUAR*CH Specifies the type of CPU architecture in client workstations. You can assign these values to *CPUARCH*:

```
R3000
R4000
```

Set *CPUARCH* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings. Check the samples in Example 3-3 (or the *client.dat* file) for the correct format of your settings.

MACH Specifies the machine architecture of the client workstations. You can assign these values to *MACH*:

```
IP6
IP12
IP20
IP22
```

Set *MACH* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings. Check the samples in Example 3-3 (or the *client.dat* file) for the correct format of your settings.

GFXBOARD Specifies the type of graphics subsystems in client workstations. A subgroup designation is also required in this setting. You can assign these values to *GFXBOARD*:

```
ECLIPSE  -mSUBGR=ECLIPSE
EXPRESS  -mSUBGR=EXPRESS
LIGHT    -mSUBGR=LIGHT
LIGHT    -mSUBGR=LG1MC
NEWPRESS -mSUBGR=NEWPRESS
NEWPORT  -mSUBGR=NG1
```

Set *GFXBOARD* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings. Check the samples in Example 3-3 (or the *client.dat* file) for the correct format of your settings.

VIDEO For Indy workstations, specifies the video chip. The default setting is null, which indicates non-Indy systems.

Set *VIDEO* to *VINO* if your client base includes any Indy workstations.

Executing the Share Tree File

The *share.dat* file and its working copies are executable shell scripts. After you edit the file, check its syntax by executing it, as shown in this example:

```
# sh IRIX_51.dat
```

If a prompt is returned after your entry, the syntax in your configuration file is correct. If you receive any other response, check your entries in the file for syntax errors.

Note: This command does not check the validity of your settings in *share.dat*; it checks file syntax only.

Installing the Share Tree

To build a share tree, you run *share_inst*, which reads the working copy of *share.dat* for configuration information; then *share_inst* automatically calls *inst* to install the software from its installation location.

At minimum, your share tree must contain the Execution Only Environments (*eoe1* and *eoe2*) and the Network File Service option (*nfs*) subsystems listed in Table 2-4. If your client users require a windowing system, you should also install the Default Window Manager (*4Dwm*), the Desktop Environment (*desktop_eoe*), Motif (*motif*), and X (*x*) subsystems listed in Table 2-4.

Hint: If you are uncertain about what to include in your share tree, accept the default subsystems selected for installation on the original software distribution (these subsystems are marked by an asterisk [*] in *inst* software listings). Add the *nfs.sw.dskless_client* subsystem to the default list. This configuration requires a considerable amount of disk space, however.

The *share_inst* Command

The *share_inst* command initiates share tree installation; it requires an argument that identifies the name of the *share.dat* file to be used in configuring the share tree. This argument is called *release_name*—it assumes that you have named your working copy of *share.dat* for the release of IRIX that it configures.

Syntax for the *share_inst* command follows (see the man page *share_inst*(1M) for a more complete description).

```
share_inst -r release_name [-d]
```

where

-r is the flag for the *release_name* argument.

release_name is the name of the working copy of *share.dat*, without the *.dat* extension. This file will provide configuration information about the share tree (see "Creating a Working Copy of the Share Tree File" on page 26 for details).

-d is the flag to delete a share tree.

The following samples illustrate *share_inst* command entries.

```
# share_inst -r IRIX_51
# share_inst -r IRIX_51B -d
```

The entry in the first sample starts the installation of a share tree called *IRIX_51*. The entry in the second sample deletes a share tree called *IRIX_51B*.

The Share Tree Installation Procedure

The procedure for building a share tree varies depending on the location of the software to be installed. By default, *inst* expects to find the installation software on a local CD; but you can also specify a remote location, such as a directory or CD-ROM drive on a different network host (the examples show a remote directory). The location of the installation software will affect the way in which you do Steps 3 and 7 of this procedure.

Note: If you are updating an existing share tree, do not use this procedure; use the procedures in Chapter 5 instead.

1. Start *share_inst* as shown in this example:

   ```
   # cd /var/boot
   # ./share_inst -r IRIX_51
   ```

   This entry directs *share_inst* to read *IRIX_51.dat* (the working copy of *share.dat*) to build a *share* tree for IRIX 5.1 clients. After it processes your entry, *share_inst* displays a confirmation prompt:

   ```
   About to install shared tree
   at/diskless/share/IRIX_51......

   Enter confirmation (y/Y):
   ```

2. Enter **y** or **Y** (yes) to confirm the start of installation.

   After your entry, the Inst Main menu is displayed:

   ```
   Default location  of new software is <previous installation location>
   If you are a first-time inst user, give the command “help beginner".
   Inst Main Menu
   1. from [source]            Specify location of software to be installed
   2. list [keywords][names]   Display information about software subsystems
   3. go                       Perform software installation and removal now
   4. install [keywords][names] Select subsystems to be installed
   5. remove [keywords][names]  Select subsystems to be removed
   6. keep [keywords][names]   Do not install or remove these subsystems
   7. step [keywords][names]   Enter interactive mode for install/remove/keep
   8. versions [args]          Get information about installed software
   9. help [topic]             Get help in general or on a specific word
   10. admin                   Go to the Administrative Commands Menu
   11. quit                    Terminate software installation

   Inst>
   ```

3. Specify the location of your software.

   If you are installing the software locally from a CD, you can accept the default location for the software:

   - Insert the CD in its drive.
   - Press **<Return>**.

   If you are installing the software from a remote source, specify its location:

   - Enter **from** to select an alternative source
   - Specify the location of the software, as shown in the following examples. These examples show entries that specify a remote directory and CD as the source of the software.

     ```
     Inst> from bigsgi:/d2/irix51
     Inst> from bigsgi:/CDROM/dist
     ```

   After your entry, *inst* confirms that it will look for the software at the location you specified; then it lists the products available at that location as shown in this example:

   ```
   Reading product descriptor from source_specification

   4Dwm        4Dwm -- Default Window Manager (...)
   motif_dev   IRIX IM Development Software (...)
   motif_eoe   IRIX IM Execution Only Environment (...)
   il_dev      ImageVision Library, 2.2
   il_eoe      ImageVision Library Execution Only ...
   insight     IRIS InSight Viewer, Release 2.2
   ```

   Hint: If you know the subsystems that you want to install and do not need to check space requirements, you can skip Step 4 and go to Step 5.

4. Enter **list** to check selected subsystems and space requirements:

   ```
   Inst> list
   Computing disk space changes:
   ```

After your entry, the *list* option calculates the disk space needed to install the selected subsystems. After several minutes, it posts a list of available subsystems similar to the partial listing in this example:

```
Subsystem                                    /Description

eoe1.man.relnotes * 121+ Workstation Release Notes
eoe1.man.slip *      21+ SLIP Manual Page
eoe1.man.unix *    6272+ Basic IRIX Documentation
eoe1.sw.audio *    3809+ Audio Software
eoe1.sw.cpp *@      119+ Source Code Preprocessor
eoe1.sw.lboot *@+  6627+ Kernel lboot software
eoe1.sw.lib *@+   31801+ Execution C/Fortran Library
eoe1.sw.quotas @    259+ BSD Disk Quotas

More?
```

To advance through the list, press the space bar or **`<Enter>`** at the `More?` prompt. At the end of the list, you see the `Inst>` prompt.

5. Enter **`step`** to select software subsystems for the share tree:

   ```
   Inst> step
   ```

   After your entry, the first item on the list of subsystems in the distribution is displayed. If you are not sure which subsystems to include in the share tree, see Table 2-4 for a suggested minmum.

   Note: You must select all *eoe1*, *eoe2*, and *nfs* subsystems listed in Table 2-4 for installation.

   To modify a selection on the *inst* list , type one of these entries:

   `<Enter>` to leave the selection unchanged.

   `r <Enter>` if it is marked *i* and you do not want it installed

   `i <Enter>` if it is not marked *i* and you want to install it

   When the list is finished, you see a report of your changes; then the `Inst>` prompt is displayed.

6. Check your list of selections (optional).

   Since it is time consuming to install a share tree, you might want to verify your selections by typing **`list`** again.

7. Enter **go** to launch the installation.

   ```
   Inst> go
   ```

   ```
   Installing new versions of selected subsystem_name
   subsystems.
   ```

   As it works, *inst* displays messages that identify the subsystems being installed. You see this message when the installation is finished:

   ```
   Installation and/or removal succeeded.
   You can insert another tape or CD-ROM now.
   ```

   ```
   Inst>
   ```

8. Insert the next CD, if needed.

   If you have no more software to install, go to Step 9. If you have more CDs to install, do this:

   - Insert the new CD.
   - Press **<Enter>**.
   - Repeat Steps 5 through 7 for each additional CD.

   Note: If your new share tree is to contain an IRIX maintenance release, insert the CD containing the maintenance release now. Repeat Steps 4 through 7 to install your maintenance software.

9. Enter **quit** to complete the installation:

   ```
   Inst> quit
   ```

   ```
   Running exit commands...
   ```

   If there is a problem with the *exports* file, you will see a message in your current window and a message will be sent to the console. If there is a problem in *exports*, correct the */etc/exports* file; then run *exportfs* **-a** manually to correct the problem.

Allow the *exit* commands several minutes to process.

Caution: Do not make any entries that will interrupt processing of *exit* commands (such as **<Ctrl>** C, for example). Doing so can result in damage to your system.

When processing is complete, the share tree is installed on the server system. To verify the installation, change to the directory for the share tree and list its contents:

```
# cd /diskless/share/IRIX_51
# ls
dev/ lib/ sbin/ var/
```

Chapter 4

Creating Client Trees

To install software in a client tree, you create a working copy of the client tree configuration file, */var/boot/client.dat*, and edit the working copy to specify the requirements of your client and swap trees. Then you run the client tree installation script, */var/boot/client_inst*, which installs individual client and swap trees using the specifications in your working copies of *client.dat* and *share.dat*. After the client and swap trees are installed, you can boot the client workstation and put it into service.

This chapter contains these sections:

- "The Client Tree Configuration File"
- "Creating a Working Copy of the Client Tree File"
- "Editing the Client Tree File"
- "Executing the Client Tree File"
- "Installing the Client and Swap Trees"
- "Booting the Client Workstation"

The Client Tree Configuration File

The configuration file for a client tree is a customized copy of the */var/boot/client.dat* file. This file defines the root directory for client tree and swap tree, the size of the swap space, and the hardware components of the client workstation using the tree.

Although each client in a diskless implementation must have its own client tree, it is not necessary to create an individual working copy of *client.dat* for each client. Since the information in a working copy of *client.dat* frequently pertains to an entire client class, a single copy of the file can often be used to create multiple client trees.

Figure 4-1 illustrates the effects of using a single copy of *client.dat* to create multiple client trees.

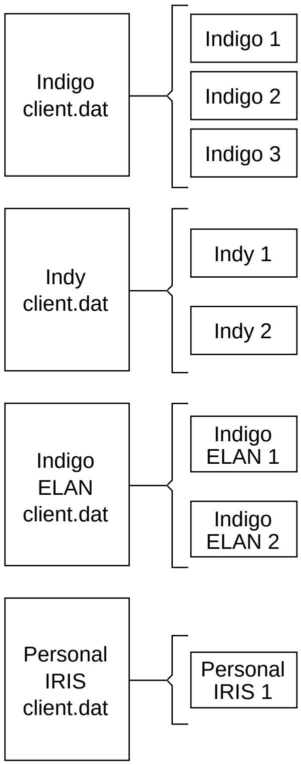

Figure 4-1 Multiple Client Trees from One Configuration File

You can use the same copy of *client.dat* for multiple client trees whenever these conditions are true:

- Clients are identical workstation models (including graphics)
- Client trees for all clients will be in the same directory
- Swap trees for all clients will be in the same directory and contain the same amount of swap space.

For example, a single working copy of *client.dat* can be used to install the client trees for a group of Indigo R4K Elans with client trees in the

/diskless/client directory, swap trees in the */diskless/swap* directory, and 20 Mb of swap space.

Creating a Working Copy of the Client Tree File

Your server should already have a distribution copy of *client.dat* in the */var/boot* directory as a result of installing Diskless Workstation software. Do not edit the distribution copy of *client.dat* ; leave the distribution copy in its original form and make working copies of the file to enter your changes.

By convention, a working copy of *client.dat* is named for the workstation model that it configures, and it ends with a *.dat* extension. The following examples illustrate command lines that create and name a working copy of *client.dat* for Indigo, Indigo XS, and IRIS 4D/20 clients:

```
# cp /var/boot/client.dat /var/boot/Indigo.dat
# cp /var/boot/client.dat /var/boot/Indigo_XS.dat
# cp /var/boot/client.dat /var/boot/4D_20.dat
```

Editing the Client Tree File

The variables in the distribution copy of *client.dat* are assigned default values. You will need to change many of these values to configure the client trees for your diskless implementation.

The subsections that follow illustrate each section of *client.dat* . Below each illustration is an explanation of variables in the section and recommendations for setting them. Use these subsections as a guide when making your working copies of *client.dat*.

Setting Client and Swap Tree Variables

The variables in the upper portion of *client.dat* specify the location of the client tree and swap tree, the size of the swap space, and the NIS domain name for clients using this client tree configuration file.

Example 4-1 illustrates the upper portion of the *client.dat* file.

Example 4-1 The Client Tree Setup Section of *client.dat*

```
# Information for diskless client setup.
# Please make a copy of this file for each class of diskless
machines.
#
# $Revision: 1.5 $
#
#
# CLIENT TREE SETUP
#
#      CLROOT :    directory where client tree resides.
#      SWAP :      directory where swap file resides.
#      SWAPSIZE :  size of swap file.
#                  Same format as "mkfile" takes.
#      NISDOMAIN : If set to any value, then NIS domain is
set
#                  as specified. If set to null, then
client's
#                  NIS domain is set as diskless server's
NIS domain
#                  unless client's NIS domain is set
previously.
#
#      NOTE:
#             If one wants to prevent the client from using
#             swap space on server, then set SWAPSIZE="0".
#             However, If one still wants to preserve already
#             existing swap space on server for some reason,
#             then set SWAP="".
#
CLROOT="$DISKLESS/client/$HOST"
SWAP="$DISKLESS/swap/$HOST"
SWAPSIZE="20m"
NISDOMAIN=" "
```

The variables in this portion of *client.dat* have these meanings:

CLROOT Specifies the pathname of the client tree (see Figure 2-1). The default value is $*DISKLESS/client/$HOST*

Reset this value if you plan to install your client trees in a directory other than the one specified by the *$DISKLESS* variable; be sure your entry ends in the string *$HOST* The value of *$HOST* is the host name of the client that you enter on the *client_inst* command line to install the client tree.

SWAP Specifies the pathname of the swap tree (see Figure 2-1). The default value is *$DISKLESS/swap/$HOST*.

Reset this value if you plan to install your swap trees in a directory other than the one specified by the *$DISKLESS* variable; be sure your entry ends in the string *$HOST*. The value of *$HOST* is the host name of the client that you enter on the *client_inst* command line to install the swap tree.

SWAPSIZE Specifies the size of the swap space, in megabytes. The default value is 20 megabytes. If you change this value, your entry must end with the letter *m* (lower case), the only allowable value for megabytes.

NISDOMAIN Specifies the domain name for a client. The default setting is null (" "), which assigns the server's domain name to its clients. This setting can vary for individual clients in a class.

Set *NISDOMAIN* to null (" ") to assign the server's domain name to the client. The *client_inst* script will enter the server's domain name in the client's copy of */var/yp/ypdomain*.

Set *NISDOMAIN* to an alternative domain name if clients are using a domain name that is different from the server's. The *client_inst* script will enter the domain name you specify in the client's copy of */var/yp/ypdomain*.

Note: You will need to set *yp* to *on* with *chkconfig*(1M) on the client so that the client can participate in NIS.

Setting Client Architecture Variables

The variables in the lower portion of *client.dat* specify the hardware components of client workstations.

Example 4-2 illustrates the table in the lower portion of *client.dat*.

Example 4-2 Architecture Tables in *client.dat*

```
# Client CPUARCH/CPUBOARD/MACH/GFXBOARD values for IRIX releases 5.1 and
# beyond. CPUARCH should be omitted for releases prior to 4.0.4, but will
# not hurt if present (as long as the server is at 4.0.4 or later)
#
# NOTE    NOTE    NOTE    NOTE    NOTE
# CPUARCH should be set to MIPS1, rather than R3000, or MIPS2, rather than
# R4000 if the *client* (diskless) machine will have IRIX 4.0.4 or any of
# IRIX 4.0.5[ABC] installed on it. The values shown below are correct
# for clients that will run IRIX 4.0.5D or later.
#
# Key: 4D/20{G,TG} is an abbreviation for 4D/20, 4D/20G, and 4D/20TG.
#+-------------------+---------+---------+------------------------------------+
#| Machine Type      | CPUARCH | CPUBOARD| GFXBOARD                  | VIDEO  |
#|                   |         | /MACH   |                           |        |
#+-------------------+---------+---------+---------------------------+--------|
#| 4D/20{G,TG}       | R3000   | IP6     | "ECLIPSE -mSUBGR=ECLIPSE" |        |
#| 4D/25{G,TG}       | R3000   | IP6     | "ECLIPSE -mSUBGR=ECLIPSE" |        |
#|                   |         |         |                           |        |
#| 4D/30{G,TG}       | R3000   | IP12    | "ECLIPSE -mSUBGR=ECLIPSE" |        |
#| 4D/30EG (Elan)    | R3000   | IP12    | "EXPRESS -mSUBGR=EXPRESS" |        |
#|                   |         |         |                           |        |
#| 4D/35{G,TG}       | R3000   | IP12    | "ECLIPSE -mSUBGR=ECLIPSE" |        |
#| 4D/35EG (Elan)    | R3000   | IP12    | "EXPRESS -mSUBGR=EXPRESS" |        |
#|                   |         |         |                           |        |
#| Indigo            | R3000   | IP12    | "LIGHT -mSUBGR=LIGHT"     |        |
#| Indigo Dual Head  | R3000   | IP12    | "LIGHT -mSUBGR=LIGHT"     |        |
#| Indigo XS{24}     | R3000   | IP12    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Indigo Elan       | R3000   | IP12    | "EXPRESS -mSUBGR=EXPRESS" |        |
#|                   |         |         |                           |        |
#| Indigo R4K        | R4000   | IP20    | "LIGHT -mSUBGR=LG1MC"     |        |
#| Indigo R4K DH     | R4000   | IP20    | "LIGHT -mSUBGR=LG1MC"     |        |
#| Indigo R4K XS     | R4000   | IP20    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Indigo R4K XZ     | R4000   | IP20    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Indigo R4K Elan   | R4000   | IP20    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Note: Indigo is also known as 4D/RPC. |                           |        |
#|                   |         |         |                           |        |
#| Indigo2 Extreme   | R4000   | IP22    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Indigo2 XZ        | R4000   | IP22    | "EXPRESS -mSUBGR=EXPRESS" |        |
#| Indigo2 Extreme+XL| R4000   | IP22    |"NEWPRESS -mSUBGR=NEWPRESS"|        |
#| Indigo2 XL        | R4000   | IP22    | "NEWPORT -mSUBGR=NG1      |        |
#|                   |         |         |                           |        |
#| Indy              | R4000   | IP22    | "NEWPORT -mSUBGR=NG1"      | "VINO" |
#+-------------------+---------+---------+----------------------------+--------+
#
CPUBOARD="IP20"
CPUARCH=R4000
MACH="IP20"
GFXBOARD="LIGHT -mSUBGR=LG1MC"
VIDEO=" "
```

The variables in the hardware specification section of *client.dat* have these meanings:

CPUARCH Specifies the type of CPU architecture in a client workstation. The default value for *CPUARCH* is *R4000*.

Set *CPUARCH* to the value for the workstation model using this client tree. *CPUARCH* must be assigned only one value in *client.dat*. See Example 4-2 or the tables in *client.dat* for valid entries.

CPUBOARD Specifies the CPU subsystem in a client workstation. The default value for *CPUBOARD* is *IP20*.

Set *CPUBOARD* to the workstation model using this client tree. *CPUBOARD* must be assigned only one value in *client.dat*. See Example 4-2 or the tables in *client.dat* for valid entries.

MACH Specifies the client's CPU type to selected software modules. The default value for *MACH* is *IP20*.

Set *MACH* to the value for the workstation model using this client tree. *MACH* must be assigned only one value in *client.dat*. See Example 4-2 or the tables in *client.dat* for valid entries.

GFXBOARD Specifies the type of graphics subsystem in a client workstation. A subgroup designation is also required in this setting. The default value for *GFXBOARD* is *LIGHT,* whose subgroup is *LG1MC*.

Set *GFXBOARD* to the value for the workstation model using this client tree. *GFXBOARD* and its subgroup must be assigned only one value in *client.dat*. See Example 4-2 or the tables in *client.dat* for valid entries.

VIDEO For Indy workstations, specifies the video chip. The default setting is null, which indicates non-Indy systems.

Set *VIDEO* to *VINO* if your client is an Indy workstation; if it is not an Indy, use the default setting.

Executing the Client Tree File

The *client.dat* file and its working copies are executable shell scripts. After you edit the file, check its syntax by executing it, as shown in this example:

```
# sh Indigo_XS.dat
```

If a prompt is returned after your entry, then the syntax in your configuration file is correct. If you receive any other response, the execution was not successful. Check your entries in the working file and correct any syntax errors. Then, execute the working file again.

Note: This command does not check the validity of your settings in *client.dat*; it checks only the file syntax.

Installing the Client and Swap Trees

To install a client tree, you run *client_inst*, which reads the working copy of *client.dat* and *share.dat* for configuration information; then *client_inst* calls *inst* to install the software. Since *inst* uses the share tree as its software source, you must have installed the share tree before installing any of the client trees that will use it. Unless you specify otherwise, *client_inst* installs all software subsystems in the share tree that are marked as the default selections on *inst* listings.

Note: You must install the *nfs.sw.dskless_client* subsystem in every client tree.

As part of the installation process, *client_inst* installs the client's swap tree. It also creates the client's */etc/fstab* and */etc/hosts* files and includes them in the client tree. Finally, *client_inst* adds entries for the client in the server's */etc/exports* file and */etc/bootparams* file (or that of the NIS master server). The internet address for a client must be available to network software when you install its client tree(see Chapter 2, “Assigning Internet Addresses” for details).

Caution: Do not attempt to install client trees simultaneously by running concurrent *client_inst* processes. Concurrent *client_inst* processes can damage the server's */etc/exports* and */etc/bootparams* files.

The *client_inst* Command

The *client_inst* command initiates client and swap tree installation. *client_inst* requires arguments that identify the host name of the client being installed, and the working copies of *share.dat* and *client.dat* to be used in configuring the client tree.

Syntax for the *client_inst* command is shown below (see the man page *client_inst* (1M) for a more complete description).

Note: This syntax does not include the *clone* option, which is described in “Cloning Client Trees”

```
client_inst -h hostname -r release -c class [-d]
```

where

-h is the flag to specify the host name of a client

hostname is the host name of the client whose tree you are installing

-r is the flag to specify a share tree (release)

release is the working copy of *share.dat* that defines this client's share tree (without the *.dat* extension)

-c is the flag to specify a class

class is the name of the working copy of *client.dat* for this class (without the *.dat* extension)

-d is the flag to delete a client tree

The following samples illustrate *client_inst* command entries.

```
client_inst -h starlite -r IRIX_51 -c Indigo_XS
client_inst -h starlite -r IRIX_51 -c Indigo_XS -d
```

The entry in the first sample installs a client tree and swap tree for a client named *starlite*, whose client tree is defined in a file named *Indigo_XS* (the *.dat* extension is omitted) and whose share tree is defined in the file called *IRIX_51* (the *.dat* extension is omitted). The entry in the second sample deletes the client tree for host *starlite*.

The Client and Swap Tree Installation Procedure

When you install a group of clients, you can either run *client_inst* separately to install each client tree, or you can install one client tree and clone the tree for additional clients (see “Cloning Client Trees”). A client's swap tree is automatically installed when you install or clone its client tree.

Note: If you are updating an existing client tree, do not use the procedures that follows. Use the update procedures in Chapter 5 instead.

1. Start *client_inst* as shown in this example:

   ```
   # cd /var/boot
   # ./client_inst -h starlite -r IRIX_51 -c Indigo_XS
   ```

   This entry directs *client_inst* to create a client tree for *starlite*, using configuration information in *IRIX_51.dat* for its share tree and *Indigo_XS.dat* for its client tree. After processing your entry, *client_inst* displays this prompt:

   ```
   Client tree = /diskless/client/starlite,
   shared tree = /diskless/share/IRIX_51
   Enter confirmation (y/Y):
   Create 20m swap file ........
   ```

2. Enter **y** or **Y** (yes) to confirm the start of installation.

   After your entry, the Inst Main menu is displayed:

```
Default location of new software is <previous installation location>
If you are a first-time inst user, give the command "help beginner".
Inst Main Menu
1. from [source]             Specify location of software to be installed
2. list [keywords][names]    Display information about software subsystems
3. go                        Perform software installation and removal now
4. install [keywords][names] Select subsystems to be installed
5. remove [keywords][names]  Select subsystems to be removed
6. keep [keywords][names]    Do not install or remove these subsystems
7. step [keywords][names]    Enter interactive mode for install/remove/keep
```

```
8. versions [args]           Get information about installed software
9. help [topic]              Get help in general or on a specific word
10. admin                    Go to the Administrative Commands Menu
11. quit                     Terminate software installation&lbreak;
Inst>
```

3. Enter **`list`** to check your share tree subsystems (optional).

   The *list* option displays the contents of your share tree and the subsystems selected for installation. This is the default set of subsystems for a client tree, and they are marked by the letter *i*.

   The *list* option also calculates the disk space needed to install the selected options and displays the amount of disk space that is available for installation. After completing its calculations, *list* posts a disk space message like the one shown in this example:

   ```
   Net change in blocks 4538+
   Blocks available 795668+
   ```

4. Install the *nfs.sw.dskless_client* subsystem.

   This subsystem is required for diskless operation.

   ```
   Inst> install nfs.sw.dskless_client
   ```

5. Select the other subsystems that you want to install.

   If you want to accept the default selections for this client tree, go to Step 5. If you want to change any of the defaults, use the *step* option to make your selections (for details, see Step 5 of "The Share Tree Installation Procedure" in Chapter 3).

   Hint: You can use two alternatives to the *step* option to make your selections. The *install* option lets you specify individual subsystems that you want to include in the client tree (your selections must be installed on the share tree), and the *remove* option lets you specify individual subsystems that you want to omit from the client tree.

6. Enter **go** to launch the installation:

   The *go* option installs all software subsystems marked for installation.

   As it works, *inst* reports status messages and identifies subsystems that it is installing.

   ```
   Installing new versions of selected subsystem.
   ```

   When it finishes, *inst* displays this message:

```
Installation and/or removal succeeded.
You can insert another tape or CD-ROM now.
```

The message indicates that client tree software is successfully installed.

Enter **quit** to finish the installation:

After your entry, *inst* processes a suite of *exit* commands. Allow the *exit* commands several minutes to process.

Caution: Do not make any entries that will interrupt processing of the *exit* commands (such as **<Ctrl> C**, for example). Doing so can result in damage to your system.

When processing is complete, one client tree is installed on the server system. Repeat this procedure for each diskless workstation that you plan to install.

Cloning Client Trees

Once you have installed one client tree, you can use the installed tree as a model for creating additional client trees with *-clone*, an option to the *client_inst* command. Like the conventional *client_inst* process, the clone option automatically creates a swap tree when it creates a client tree.

Cloning client trees takes less time than conventional installations, since the clone option bypasses *inst*. You can save additional time by customizing the private files on the model client tree before you clone it. For example, you could edit the */etc/fstab* on the model tree to include the mount entries that all cloned trees require; you could do a similar edit on the model tree's copy of */etc/hosts* and */etc/resolv.conf*, and so on.

To clone a client tree, use this syntax of the *client_inst* command:

```
client_inst -h hostname -r release - c class -clone model
```

Specify this information in command arguments:

-h *hostname* is the host name of the client whose tree is being created by the cloning process

-r *release* is the working copy of *share.dat* that defines this client's share tree (without the *.dat* extension). The share tree for the model and all its clones must be the same.

-c *class* is the name of the working copy of *cl ient.dat* that was used for the model client tree (without the *.dat* extension).

-clone *model* is the complete pathname to the model client tree. For example, if starlite's client tree is used as the model, this argument will read *-clone /diskless/client/starlite*.

Booting the Client Workstation

Once you finish building a client tree, diskless software is ready to support clients. The procedure that follows explains how to prepare a client workstation, then boot it to run as a diskless system.

Note: If you are using the EPROM workaround on any client because it contains an early revision EPROM, you need to repeat Step 4 of this procedure each time the client is rebooted (see "Checking a Client" in Chapter 2 and "Compensating for EPROM Hardware" in Chapter 7 for details).

Go to the client workstation to do this procedure:

1. Shut down the client, if it is up.

   If the client is up, type **`shutdown`** on the console to bring down the system.

2. Go to the Command Monitor.

   This step is different for different workstation models. See the *Owner's Guide* for this model if you need instructions.

3. Check the NVRAM variables.

   Enter **`printenv`** and check its output for NVRAM variables. It should have several lines of these variables.

4. Set the NVRAM variables that support diskless operation.

The sample commands that follow illustrate how to set NVRAM variables. Notice that Indigo R4K models require two variables to specify the boot file setting:

```
>>setenv diskless 1
>>setenv netaddr 192.0.2.6
>>setenv bootfile bootp()sandy:/var/boot/starlite/unix
```

The variables in these examples have the following meaning:

diskless is the variable that enables diskless service. Set *diskless* to 1.

netaddr is the client's internet address. See "Assigning Internet Addresses" in Chapter 2 if you need help with this setting.

bootfile ; is the complete pathname for the client's boot file (except on R4K Indigo systems). In the sample command above, */var/boot/starlite/unix* on server *sandy* will be used to boot *starlite*.

If your client is an R4K system, use these two variables in place of the *bootfile* variable to specify boot file location:

```
>>setenv SystemPartition bootp()sandy: /var/boot/starlite
>>setenv OSLoader /unix
```

The variables in these examples have the following meaning:

SystemPartition is the directory where the client's boot file is located. In the previous sample command, the directory */var/boot/starlite* on server *sandy* contains the boot file for *starlite*.

OSLoader is the name of the IRIX boot file. The previous sample command specifies the boot file to be */unix*.

5. Enter **`printenv`** to verify your entries.

   Check the values of the NVRAM variables in the output from *printenv* to verify that the settings are correct.

6. Enter **`init`** to set NVRAM variables.

If your system's Ethernet transceiver has light emitting diodes (LED), you will see Ethernet activity for about 3 to 10 minutes, depending on other Ethernet activity. After a few minutes, the Command Monitor menu is displayed.

7. To start the client, select `1) Start System`.

Once booting is complete, the client workstation is running as a diskless workstation. Step 4 of this procedure should be repeated if any environment variables change (if the client gets a new internet address, for example); or each time you reboot this client, if you are using the EPROM workaround.

Chapter 5

Updating Software on Diskless Trees

This chapter explains how to update the software on a diskless implementation that is already running an earlier software release. It explains how to install the updated share tree and client trees, and how to create a new client tree that contains updated software. It also explains how to add software options to a share tree and client tree.

This chapter contains these sections:

- “Points to Keep in Mind”
- “Updating a Share Tree”
- “Updating Client Trees”
- “Updating New Client Trees”
- “Adding Software Options”

Points to Keep in Mind

When you use these procedures, keep software compatibility considerations in mind: the revision level of the server software supporting the diskless implementation must be equal to or greater than the software on the share tree. In addition, you must update the share tree before updating any of its client trees.

All clients should be shut down while you update their share tree. An individual client should be shut down while you update its client tree.

Note: If you do not shut down clients to perform an update, they might not operate properly after the installation.

Updating a Share Tree

The update procedure replaces the software on a share tree with a later software release. The update procedure also adds new subsystems to the share tree and removes subsystems that are no longer needed.

Since software was previously installed in the tree you are updating, *inst* has a record of the installation; it uses the record to identify currently installed software during the update process. In any software listings that it displays, *inst* marks currently installed subsystems with the letter *I* in the leftmost column. In cases where the new software contains an updated version of these subsystems, *inst* replaces the subsystem with the newer version.

The following procedure explains how to update software on a share tree:

1. Shut down all client workstations in this class.
2. Change to the */var/boot* directory:

   ```
   # cd /var/boot
   ```

3. Enter the *share_inst* command, substituting the name of your share tree for the one shown in this example:

   ```
   # ./share_inst -r IRIX_51
   ```

   Note: If you have not shut down the clients using this share tree, *share_inst* displays this message:

   ```
   WARNING: release IRIX_51 is still serving clients.
   continue to update IRIX_51(Y/N)?
   ```

   If you answer **Y** (yes), *share_inst* posts a final warning:

   ```
   Checking clients status:
   client client_1 ... UP
   WARNING: clients will be out of sync. Clients which
   are UP may crash.
   ```

   If clients have been shut down (or if you proceed with the installation in spite of the warnings), *share_inst* displays a confirmation prompt:

   ```
   About to install shared tree at
   /diskless/share/IRIX_51......
   ```

   Enter confirmation (y/Y):

4. Enter **y** or **Y** (yes) to confirm the installation.

   After your entry, the Inst Main Menu is displayed:

```
Default location of new software is <previous installation location>
If you are a first-time inst user, give the command "help beginner".
Inst Main Menu
from [source]            Specify location of new softwareto be installed
list [keywords] [names]    Display information about software subsystems
go                       Perform software installation and removal now
install [keywords] [names] Select subsystems to be installed
remove [keywords] [names]  Select subsystems to be removed
keep [keywords] [names]    Do not install or remove these subsystems
step [keywords] [names]    Enter interactive mode for install/remove/keep
versions                 Get information about installed software
help [topic]             Get help in general or on a specific word
admin                    Go to the Administrative Commands Menu
quit                     Terminate software installation

Inst>
```

5. Enter **`from`** to specify the software location.

   The *inst* utility assumes that the location of your update software is the same as it was for the previous installation. (Notice the “default location” message above the Main menu.)

   - If you plan to install the updates from the same location, insert the CD (if needed) and press **`<Enter>`**; then go to Step 6.
   - If you plan to install the updates from a location other than the default, specify the location with the *from* option. Insert the CD (if needed) and press **`<Enter>`**; then go to Step 6.

6. Enter **`list`** to determine what is currently installed (optional).

   Use the *list* option if you want to check the current installation in this share tree before installing the updates. Installed subsystems are marked with the letter *I*, unless the update medium contains a newer version. If there is an update for a subsystem, it is marked with the letter *K*.

7. Enter **`step`** to select the subsystems that you want to install.

   Use the *step* option to select the subsystems that you want installed to update this share tree. (See Step 5 of "The Share Tree Installation Procedure" in Chapter 3 or on-line help if you need instructions for using *step*.)

   Hint: You can also use two alternatives to the *step* option to make your selections. Use the *install* option to specify individual subsystems that you want to update or add to the share tree (your selections must be contained on the update medium); or, use the *remove* option to specify individual subsystems that you want to omit from the updated share tree.

8. Enter **go** to launch the installation.

   As your share tree software is updated, you will see messages such as the examples shown below.

   ```
   Removing old versions of selected maint1.eoe1_sw
   subsystems
   Installing new versions of selected maint1.eoe1_sw
   subsystems

   Removing old versions of selected maint1.eoe1_man
   subsystems
   Installing new versions of selected maint1.eoe1_man
   subsystems

   Removing old versions of selected maint2.eoe2_sw
   subsystems
   Installing new versions of selected maint2.eoe2_sw
   subsystems

   Removing new versions of selected maint2.eoe2_man
   subsystems
   Installing new versions of selected maint2.eoe2_man
   subsystems
   ```

   At the end of the installation, you will see this message:

   ```
   Installation and/or removal succeeded.
   You can add another tape or CD-ROM now.
   ```

9. Enter **`quit`** to complete the update procedure.

   Allow the *exit* commands several minutes to process.

   Caution: Do not make any entries that will interrupt *exit* command processing (such as **`<Ctrl>`** `C`, for example). Doing so can result in damage to your system.

When processing is complete, the share tree for the class you specified is updated, and you are ready to update client trees for the class.

Updating Client Trees

The update procedure for client trees automatically installs updated software from the share tree into a client tree. You must complete this procedure for all client trees that you wish to update.

Note: If you are creating a new client tree and you want it to contain updated software, you must complete Step 2 of this procedure twice. (See “Updating New Client Trees” on page 62 for an explanation.)

Updating clients involves three general steps:

1. Shut down the workstation whose client tree you are updating.

   If you do not shut down the client, *client_inst* warns you that it might crash during the update.

2. Use *client_inst* to install an updated client tree.

   Complete all instructions in “The Client and Swap Tree Installation Procedure” in Chapter 4 to install an updated client tree. If this is a new client tree, complete the instructions in that procedure twice.

3. Reboot the client workstation.

   See “Booting the Client Workstation” in Chapter 4 for instructions.

Once the boot procedure is completed, the client is operating with updated software.

Updating New Client Trees

If you want to create a new tree containing updated software (that is, put a new client workstation into service running the updated software), you must run *client_inst* twice. The first time that *client_inst* invokes the *inst* program, *inst* searches the client tree for a record describing previously installed software. Since no record exists on a new client tree, *inst* installs a baseline version of Silicon Graphics software, without adding updates. The second time that *client_inst* invokes *inst*, *inst* uses the baseline software to add the updates to the new tree.

Adding Software Options

Whenever you add a software option, you must add the software to the share tree first. Then you add it to the client tree of each diskless workstation that is to use the option. Options you add to a share or client tree need not be installed on the server.

The procedures for adding an option to a share tree and client tree are given in the sections that follow.

Adding Options to a Share Tree

To add a software option to the share tree, follow these steps:

1. Log in to the server workstation as the superuser.
2. Enter the *share_inst* command to install software on a share tree.

   Your entry will look like the example that follows. In this example, IRIX_51 is the share tree to which the option is being added.

   ```
   # /var/boot/share_inst -r IRIX_51
   ```

After your entry, *share_inst* asks you for a confirmation, then the *inst* Main menu is displayed:

```
Default location of new software is <previous installation location>
If you are a first-time inst user, give the command "help beginner".
Inst Main Menu
from [source]             Specify location of new softwareto be installed
list [keywords] [names]   Display information about software subsystems
go                        Perform software installation and removal now
install [keywords] [names]  Select subsystems to be installed
remove [keywords] [names]  Select subsystems to be removed
keep [keywords] [names]   Do not install or remove these subsystems
step [keywords] [names]   Enter interactive mode for install/remove/keep
versions                  Get information about installed software
help [topic]              Get help in general or on a specific word
admin                     Go to the Administrative Commands Menu
quit                      Terminate software installation

Inst>
```

3. Enter **`from`** to specify the software location.

   The *inst* utility assumes you are installing the software option from the same location as the previous installation (the "default location" message above the Main Menu specifies CD, since no options have previously been installed on this tree). If you plan to use the default location, go to Step 4.

   If you plan to install the software from a location other than the default, use the *from* option to specify the new location. See the *inst* on-line help if you need instructions.

4. Enter **`keep *`** to clear unwanted selections.

   Software at the installation location is likely to have many subsystems selected for installation in addition to the option you want to install. The **`keep *`** entry tells *inst* to clear all subsystems selected for installation.

   After your entry, *inst* identifies the software at the installation location and posts a list such as the partial listing shown in this example:

```
Reading product description from guest@bigsgi:/d2/irix51
```

5. Enter **install** *option_name* to select the software option.

   For example, if you want to install the EFast™ product option, your entry would look like this:

   ```
   inst> install efast
   ```

6. Enter **list** to verify the software that you selected.

   ```
   Inst> list
   ```

7. Enter **go** to launch the installation.

   ```
   Inst> go
   ```

8. Enter **quit** to finish the installation.

   ```
   Inst> quit
   ```

When processing is complete, the option is added to the share tree. The option can now be added to client trees that will be using it.

Adding Options to a Client Tree

This section describes the procedure for adding a software option to a client tree. The following steps must be completed for each client using the new option. The option must already be installed on the share tree when you do this procedure.

1. Enter the *client_inst* command to add software to a client tree.

   ```
   /var/boot/client_inst -r IRIX_51 -c Indigo_XS -h starlite
   ```

   In this example, *starlite*, the client using the IRIX_51 share tree, is receiving the software option. After your entry, *client_inst* asks you for a confirmation, then the Inst Main menu is displayed.

2. Enter **keep *** to clear unwanted subsystem selections.

   Since many subsystems in the share tree are selected for installation, you need to clear these selections before going on with the installation. The **keep *** entry clears all selections from the share tree.

3. Enter **install** to select the software that you want to add.

   ```
   Inst> install efast
   ```

4. Enter **`list`** to verify the software that you selected.

   ```
   Inst> list
   ```

5. Enter **`go`** to launch the installation.

   ```
   inst> go
   ```

   After your entry, *inst* computes disk space changes, then installs the software option.

6. Enter **`quit`** to finish the installation.

   ```
   Inst> quit
   ```

When processing is complete, the option is added to the client tree and is available to users of this diskless workstation.

Chapter 6

Maintaining Diskless Implementations

This chapter contains procedures that are commonly required to maintain a diskless software implementation. The chapter contains the following subsections:

- "Removing a Client Tree"
- "Removing a Share Tree"
- "Changing the Size of the Swap Space"
- "Setting Up a Local Disk"
- "Moving a Client to a Different Server"
- "Converting a Client to a Standalone Workstation"

Removing a Client Tree

To remove a client tree, you use the *client_inst* command with the **-d** (delete) option. All client trees should be removed from the server before you remove their share tree. Before removing a client tree, be sure to back up anything on the tree that you want to save.

The procedure for removing a client tree is given below.

1. Enter **client_inst** with the **-d** option.

   Use the *client_inst* command with the delete option, as shown in this sample entry, which removes host *starlite*:

   ```
   # /var/boot/client_inst -h starlite -r IRIX_51 -c Indigo_XS -d
   ```

After your entry, you see a response similar to the one that follows:

```
Remove client tree at /diskless/client/starlite
                        (shared tree = /diskless/share/IRIX_51
Enter confirmation (y/Y) :y
```

2. Enter **Y** to confirm removal.

   After your entry, you see this prompt:

   ```
   Do you want host /diskless/client/starlite removed from NIS (y/n)? n
   ```

3. Enter **y** or **n** to this prompt.

   Unless you are moving this client to another network, answer **no** to this prompt. A *no* entry maintains the client's name and address in the current network's host database. If the client is moving to a new network, answer **yes**; its host name and internet address will be removed from the host database for this network.

The client tree that you specified is now removed from the server.

Removing a Share Tree

The procedure for removing a share tree is similar to the procedure for removing a client tree: you use the *share_inst* command with the **-d** (delete) option. When you use this procedure, all client trees for the class should be removed, since removing a share tree disables its clients. You should also make a backup copy of the share tree before starting this procedure.

The procedure for removing a share tree is as follows:

1. Verify that the share tree is not supporting any clients.
2. Enter **share_inst** with the **-d** option.

   Use the *share_inst* command with the delete option, as shown in the following example. If any client is still using the tree when you enter this command, *share_inst* posts warning messages (the note is shown in Step 3).

   ```
   # /var/boot/client_inst -r IRIX_51 -d
   About to remove shared tree at /diskless/share/IRIX_51....
   Enter confirmation (y/Y):
   ```

3. Enter **y** or **Y** (yes) to confirm the removal.

   After your entry, it takes several minutes for the share tree to be removed.

   Note: If you have not shut down clients for this share tree, *share_inst* displays this message after your command entry:

   ```
   WARNING: Class IRIX_51 is still serving clients.
   continue to delete IRIX_51(Y/N)?
   ```

   If you answer **y** (yes), *share_inst* posts a final warning:

   ```
   Checking clients status:
   client client_1 ... UP
   WARNING: clients will be out of sync.
   Clients which are UP may crash.
   ```

   If you proceed with the installation in spite of these warnings, *share_inst* removes the share tree.

The share tree for this version of IRIX is now removed from the server. Any clients that were using this share tree can no longer operate as diskless workstations.

Changing the Size of the Swap Space

The size of a client's swap space is set during the installation of its client and swap trees (see "Setting Client and Swap Tree Variables" in Chapter 4). However, if you find that the swap size for any client is not suitable, you can change it using this procedure:

1. Shut down the client.

   Caution: Allowing the client to remain in operation is likely to result in the loss of data.

2. From the server, change to the swap tree directory.

   This sample command changes to the swap directory for the client *starlite*:

   ```
   # cd /diskless/swap/starlite
   ```

3. Issue the *mkfile* command to change the swap size.

   This sample command changes the size of the swap file to 25 megabytes:

   ```
   # mkfile 25m _swap
   ```

4. Boot the client.

   The new swap space size goes into effect as soon as the boot process is completed.

Setting Up a Local Disk

If a client workstations contains a disk, the disk can be set up to provide local resources to the client. The procedures in this section explain how to set up a local filesystem (*/var/tmp*) and a local swap partition on the installed disk.

The procedures in this section make these assumptions:

- The local disk is properly installed in the client workstation.

  If you have not yet installed or mounted the local disk, see the *Owner's Guide* to your system for installation instructions.

- The local disk is partitioned as a system disk.

  If your disk is formatted as an option disk rather than a system disk, you can configure a filesystem on partition 7 of your disk; however, if you plan to add a swap partition to an option disk, you must first repartition it. See *Advanced Site and Server Administration Guide* and *fx*(7M) man page for instructions on disk repartitioning.

Getting Prepared

To prepare for setting up resources on the local disk, you should verify that the disk is properly installed. You should also determine the disk partition on which the new filesystem or swap partition is to be mounted and the name of the IRIX device special file that designates the partition.

Verifying Installation

To verify proper installation, issue the *hinv* command and check the output of *hinv* for an entry that identifies the newly installed disk. If the disk is properly configured, *hinv* reports the drive and its installed location. The output in the following example shows a device installed as the first drive (unit 1) on the first SCSI controller (0).

```
# hinv
Disk Drive: unit 1 on SCSI controller 0
```

Determining the Mount Partition

A local filesystem or swap partition is made available by mounting it to a particular partition on the local disk. Disk partitions are designated by a set of device special files in the */dev/dsk* directory (see *intro*(7M) for an explanation of naming conventions for special files). For example, these files pertain to the disk reported in the sample output of *hinv* above:

```
# ls /dev/dsk

/dev/dsk/dks0d1s0   /dev/dsk/dks0d1s1  /dev/dsk/dks0d1s6
/dev/dsk/dks0d1s7   /dev/dsk/dks0d1vh  /dev/dsk/dks0d1vol
```

The partition that you designate for mounting (specified by *s0*, *vh*, and so on, in the file name) will depend on whether you installed a system or option disk, and, if you have installed a system disk, on whether you are adding swap space or a filesystem to the disk. IRIX uses the conventions shown in Table 6-1 for designating disk partitions:

Table 6-1 IRIX Disk Partitions

Partition	Designation
0	*root* partition
1	*swap* partition
6	*usr* partition
7	entire disk except for the volume header
8	volume header
10	the entire disk

Note: Partition 7 is omitted from factory-shipped system disks, and partitions 0, 1, and 6 are omitted from factory-shipped option disks.

Adding a Local Filesystem

The procedure that follows explains how to set up a local */var/tmp* filesystem on a local (system) disk. It explains how to mount the filesystem, using the */etc/fstab* file in a client tree. Log in to the client workstation to do this procedure:

1. Create the mount point on the client tree:

    ```
    # mkdir /var/tmp
    ```

2. Add an entry to */etc/fstab* that mounts the filesystem.

    Open the client's copy of */etc/fstab* and add a line similar to the following sample entry (see *fstab*(4) for more information). This entry mounts the */tmp* filesystem on the root partition:

    ```
    /dev/dsk/dks0d1s0    /var/tmp    efs    rw,raw=/dev/rdsk/dks0d1s0 0 0
    ```

3. Boot the system to make your changes take effect.

When the boot is completed, the new filesystem is available to users of the client system.

Adding a Local Swap Partition

The procedure that follows explains how to set up a local swap partition on a local (system) disk using an */etc/fstab* file entry in a client tree. This swap partition will be used in conjunction with the client's swap tree, unless you remove the swap tree. When both the swap tree and a local swap partition are available, IRIX selects the swap location based on the priority setting in your */etc/fstab* entry (see *fstab*(4) for more information).

1. Make the */etc/fstab* entry for the swap space.

   Open the client's copy of */etc/fstab* and add a line similar to the following sample entry. This entry uses the standard IRIX swap partition for the local swap space. The priority setting (3) in this entry weights the swap location in favor of the local swap space:

   ```
   /dev/dsk/dks0d1s1    swap    swap    pri=3 0 0
   ```

2. Reboot the client system:

   ```
   # /etc/reboot
   ```

   The initialization sequence makes the local swap partition available for swap operations.

Moving a Client to a Different Server

When you move a client tree to a different server, be sure that the share tree on the new server is appropriate (see "Checking a Server" in Chapter 2). If you do not yet have a share tree on the new server, you must create one before doing this procedure (see Chapter 3, "Creating a Share Tree").

Moving a client to another server involves six general steps. Each step is described in detail in the sections that follow.

1. Prepare the client system.
2. Back up the old client tree.
3. Create a client tree on the new server.
4. Move private files to the new client tree.
5. Verify that the client works.
6. Clean up the old server.

Preparing the Client

To prepare the client to move to the new server, follow these steps:

1. Bring the client down gracefully.

   ```
   # shutdown
   ```

2. Power off the client and relocate it, if needed.

   If the client must be physically moved to a new location, power it off before moving it. Once it is moved, reconnect it to the new network and power it on.

3. Assign the new internet address, if needed.

   If the client is moved to a new physical network, or if it has been issued a new host name, it will need to be assigned an internet address. Obtain the internet address from your network administrator.

Backing Up the Old Client Tree

Backing up the old client tree is important for two reasons: some of the files you back up will be needed on the new server, so your copies will save you time when setting up the new client tree; the backup copies are insurance against unforeseen problems that might arise during the move.

Follow these steps to back up the old client tree:

1. Make a backup copy of the client's private files.

   Transfer the client's private files onto tape or to a temporary holding directory. You should include such files as *passwd*, *fstab*, *aliases*, and *group*. Remember that this copy of the *fstab* file will need to be modified for the new server.

2. Back up any other files the users might have requested.

   As a precaution, ask users on this client to identify any other files that reside on the client tree. Make backup copies of these files on tape or in a temporary holding directory.

Creating a Client Tree on the New Server

Once the client is moved to the new network and you have backed up files on the old server, go to the new server and create a new client tree for it:

1. Change to the */var/boot* directory.

   ```
   # cd /var/boot
   ```

2. Use *client_inst* to build the new client tree.

   ```
   # ./client_inst -r IRIX_51 -h starlite
   ```

   If you need instructions to complete this step, see "The Client and Swap Tree Installation Procedure" in Chapter 4.

Moving Private Files to the New Tree

Once the new client tree is built, move any of the client's private files to it. You can transfer the files over the network using *rcp* or copy them from tape. Follow these steps to transfer the private files with *rcp*:

1. Change to the private files'' directory.

   Log in to the new server as superuser and change to the directory that will contain the private files, as shown in this example:

   ```
   # cd /diskless/client/starlite/etc
   ```

2. Copy the private files from the old server to the new one.

   The following examples show how the files would be copied for a remote directory on the network, using *rcp*. If the private files are on a tape, use the restore command that is appropriate for your medium.

   Note: Because the filesystem on the new server might differ from that on the old, this example copies the archived version of *fstab* to a file called *fstab.old*. Creating *fstab.old* preserves the version of *fstab* made during the client tree installation.

```
# rcp oldserver:/diskless/client/starlite/etc/fstab fstab.old
# rcp oldserver:/diskless/client/starlite/etc/passwd passwd
# rcp oldserver:/diskless/client/starlite/etc/group group

# cd ../var/lib
# rcp oldserver:/diskless/client/starlite/etc/aliases aliases
```

3. Copy other files requested by users of this workstation.
4. Make a final working copy of */etc/fstab*.
5. Make additional mount points for NFS directories, if needed.

Verifying Client Operation

Once the client tree is built and the required files are transferred to the new server, verify that the client is working properly. Use this procedure to verify client operation:

1. Set the required NVRAM variables.

   The NVRAM variable *bootfile* for this client must be set to reflect the new server environment. If you changed the internet address for this client, the *netaddr* variable will also need to be changed. See Steps 4 and 5 of "Booting the Client Workstation" in Chapter 4 if you need more details.
2. Enter **`init`** and boot the client.

   This step sets the NVRAM variables and starts up the workstation using the new NVRAM values. After your entry, you are asked if you want to reconfigure the kernel.
3. Enter **`y`** to reconfigure the kernel.
4. Log in as the superuser and reboot the client.

Cleaning the Old Server

Once the client is set up and running properly on the new server, you can clean up the old one. Use the delete option (**-d**) of the *client_inst* command to remove the client tree on the old server.

Converting a Client to a Standalone Workstation

Eventually, you might decide to convert your diskless client to a standalone workstation. The procedure in this section explains how to convert a diskless client into a workstation that uses its local disk.

Converting a diskless workstation involves four general steps:

1. Install the new hardware in the workstation.
2. Enable network software on the new host.
3. Move private and working files from the server to the new host.
4. Remove the client tree from the server.

Each of these steps is described in detail in the sections that follow.

Installing the Hardware

The following instructions explain how to configure a disk drive. They assume that a fully functioning operating system is installed on the disk.

1. Follow all instructions in “Preparing the Client” on page 74.
2. Install the disk drive in the workstation.

   Follow the installation instructions sent with the disk.
3. Power on the workstation.

   After the power-on diagnostics run, you see this message:

   ```
   Starting up the system...
   To perform System Maintenance instead, press <Esc>
   ```
4. Press **`<Esc>`** to display the System Maintenance menu.
5. Invoke the workstation’s Command Monitor.

   The procedure for invoking the Command Monitor varies, depending on the workstation model. If you are not familiar with this procedure, check the *Owner’s Guide* shipped with the system. At the Command Monitor, this prompt is posted:

   ```
   >>
   ```
6. Verify that the disk was properly installed.

   ```
   >> hinv
   ```

The *hinv* (hardware inventory) command lists devices installed on this workstation. Look for output lines similar to the one shown here:

```
SCSI Disk: dksc(#,#)
```

Note: If the drive does not appear in the listing, it is not correctly installed. The drive must be removed and installed again.

7. Determine the internet address for this workstation.

   ```
   >> printenv
   ```

   Several variables are listed on the screen. Check for the value of the *netaddr* variable. If you want to keep the same internet address, make a note of the *netaddr* setting. You will need this information when you enable network software on this station.

8. Disable diskless operation.

   ```
   >> setenv diskless 0
   ```

9. Initialize the workstation environment.

   ```
   >> init
   ```

10. Exit the Command Monitor.

    ```
    >> exit
    ```

    After your entry, the System Maintenance menu is displayed.

11. Select option 1, “Start System.”

After it starts, this workstation is ready to function as a standalone system. The procedure that follows explains how to change the standalone system to a network host that can communicate with other network members.

Enabling Network Software

Many files that the new disk-based system will need are located on its former server. You can copy these files in one of two ways: enable the workstation’s network software so that it can communicate with its former server (and other hosts), or obtain a hard copy of the required files. Guidelines for turning on network software are given in the procedure below.

To enable network software, you need this information:

- The internet address for the workstation and old diskless server
- The host name for the workstation and old diskless server
- The NIS domain name for this workstation, if it uses NIS.

The following procedure explains how to turn on network software:

1. Assign a host name in */etc/sys_id*.

   Edit the */etc/sys_id* file and enter the host name for this system.

2. Enter the name of the new host in the hosts database.

   The name and internet address of this workstation must be entered in the network database files. If your network uses NIS, NIS servers must have a record of the new host name and address. If you do not administer NIS, notify its administrator to add the new host to the NIS service.

   If you do not have NIS software, add the names of other network hosts to this workstation, either with the System Manager tool (see the *Personal System Administration Guide*), or by editing the workstation's */etc/hosts* file. The remote hosts must also have the name and address of this workstation in their host database.

3. Turn on NIS software, if needed.

   If you are using NIS, set the NIS flag to *on* with *chkconfig*(1M). See the *NIS Administration Guide* for details.

4. Reboot the system so that your changes will take effect.

Moving Private Files to the New Host

To move the private files for this host from its former server, follow the instructions given here. These instructions explain how to complete the transfer with *rcp*. They assume that you are logged in to the new host and have appropriate permissions on it and the former server.

1. Copy the private files from the old server to the new host.

   The following examples show how the files are copied from a remote directory on the network, using *rcp*.

   Note: Because the filesystem on the new host differs from that on the old server, this example copies the archived version of the *fstab* file to a file called *fstab.old*. Creating *fstab.old* preserves the existing version of *fstab* (made during the installation).

```
# cd /etc
# rcp oldserver:/diskless/client/starlite/etc/fstab fstab.old
# rcp oldserver:/diskless/client/starlite/etc/passwd  passwd
# rcp oldserver:/diskless/client/starlite/etc/group group
# cd /var/lib
# rcp oldserver:/diskless/client/starlite/etc/aliases aliases
```

2. Copy other files requested by users of this workstation.
3. Make the final working version of *fstab*.
4. Make additional mount points for NFS directories, if needed.

When you finish these instructions, the workstation is ready to operate as an independent network host.

Removing the Client Tree

Once you have copied all the useful files from the server to the new disk, you should remove the client tree from the server. Follow these general steps:

1. Make a backup copy of the client tree from the server.

   This backup is for insurance; it might be required if unforeseen problems arise on the new host. See “Backing Up the Old Client Tree” on page 74 for instructions.

2. Remove the client tree from the server.

   See “Removing a Client Tree” on page 67 for instructions.

3. If no client workstations remain on the server, remove the share tree.

   See “Removing a Share Tree” on page 68 for instructions.

Chapter 7

Troubleshooting Diskless Installations

This chapter provides information to help you correct problems that might occur in your diskless implementation. It also explains what to do before you call the Silicon Graphics Technical Assistance Center and how to prepare for your call if you determine one is necessary.

This chapter contains these sections:

- "General Approach to Troubleshooting"
- "Compensating for EPROM Hardware"
- "Troubleshooting Checklist"
- "Installation Error Messages" in Chapter 7
- "Removing a Diskless Class Manually"
- "Debugging the Boot Process"
- "Handling Performance Problems"
- "Before You Call for Help"

General Approach to Troubleshooting

Frequently, problems with a diskless implementation are due to problems with the network on which it is implemented. If your diskless service is not working as expected, first determine whether there are network problems and correct any you find; then look at the diskless environment to correct any problems that persist.

Problems that are specific to the diskless environment are usually caused by one of these general conditions:

- The client EPROM is out of date.
- Diskless software is improperly configured.
- The boot process is not completing successfully.
- Client performance is unsatisfactory.

Compensating for EPROM Hardware

Booting problems can occur on older Silicon Graphics workstations equipped with EPROMs that were not designed for diskless operation ("Checking a Client" in Chapter 2 explains the details). These EPROMs contain software that does not support the *diskless* variable, which enables and disables diskless operation.

If you determined from the information in "Checking a Client" in Chapter 2 that a client EPROM does not support the *diskless* variable, you can use the workaround described in the following subsection either as a temporary or long-term solution. If the workaround does not provide a satisfactory solution, call your support provider about an EPROM upgrade.

EPROM Workaround

Even when a workstation's EPROM does not support the *diskless* variable, you can boot it as a diskless client by manually setting the variables needed for diskless operation: *netaddr*, *bootfile*, and *diskless*. The procedure below explains how to set these variables. However, since the values you assign the *bootfile* and *diskless* variables will not be set in NVRAM, you must perform this procedure each time you boot the client workstation.

Do this procedure in the Command Monitor at the client workstation:

1. Set the *netaddr* parameter.

   Use this command to store the network address, substituting your client's network address:

   ```
   >> setenv netaddr <net_address>
   ```

   This saves the network address in NVRAM. If *netaddr* is not saved after you set it, the system supports neither BOOTP nor TFTP, so it cannot be used for diskless operation (see "When Diskless Is Not an Option" on page 83).

2. Set the *bootfile* and *diskless* variables.

   Use these commands too set the bootfile and diskless variables, substituting your server name and client name:

   ```
   >> setenv bootfile bootp()<server>:/var/boot/<client>/unix
   >> setenv diskless 1
   ```

3. Boot the workstation.

   ```
   >> boot
   ```

   Use the *boot* command to go into the booting phase immediately, so that the values for the *bootfile* and *diskless* parameters are retained in RAM. This step is critical to files in the kernel, which use */etc/nvram* to read the *diskless 1* parameter when setting up the filesystem during kernel booting (see *nvram(1M)*).

   Note: Do not use the Command Monitor's *init* or *exit* commands instead of the *boot* command. If you use *init* or *exit*, the values you are trying to set will be cleared.

When Diskless Is Not an Option

In a few cases, booting problems occur because the EPROM does not support *bootp*, the network booting protocol. Workstations that do not support *bootp* are incapable of operating as diskless clients.   If you want to use this workstation as a diskless client, an EPROM upgrade is required. You can request an EPROM upgrade from your support provider.

Troubleshooting Checklist

Some problems you might encounter with your diskless configuration can be solved by considering these questions:

1. Are you certain you installed an official IRIX release?

   If you suspect that the software you are using is not an officially released version of IRIX (such as beta software, for example), remove and reinstall the share tree and any client trees made from the share tree.

2. Has any software in the diskless tree been modified since *share_inst* or *client_inst* installed it?

   You can answer this question with a simple test. Try to remove the client tree with *client_inst*. At the beginning of this procedure, *client_inst* checks various tables and files to verify that they are in the same form as when it built them. If *client_inst* finds anything changed, it refuses to remove the client tree.

   Note: You can safely do this test without deleting the client tree, since *client_inst* will ask you for confirmation before it removes anything on the tree.

3. Are hardware variables set correctly in *share.dat* and *client.dat*?

   Values for *CPUBOARD, CPUARCH, MACH*, *GFXBOARD*, and *VIDEO* must be set correctly during software installation and should not be changed once the share tree is built. All hardware settings in individual *client.dat* files must be represented by a corresponding setting in the *share.dat* file that defines their share tree.

4. Did you try to run *lboot* on the share tree or a client tree?

   If you ran *lboot*, remove and reinstall all software on any share and client tree where you ran this utility.

5. Are the versions of IRIX on the share tree suitable for the server?

   Although you can run any version of IRIX on a server, the share trees it supports cannot be running later software versions than the server itself. To eliminate the possibility of compatibility problems, it is recommended that the version of IRIX on share trees be the same as the version of IRIX on the server supporting them, particularly if you are running maintenance releases.

Installation Error Messages

Table 7-1 shows error messages that can be displayed when you are using *share_inst* or *client_inst* to build a diskless tree. It also suggests what you can do to correct the problems.

Table 7-1 Installation Error Messages

error messages	check here
`/var/boot/classname.dat not found`	The *client.dat* file for the share tree that *client_inst* is trying to build is missing. Check */var/boot* to make sure that the appropriate *client.dat* is listed and check your spelling of the file name.
`Invalid hostname_clientname`	*client_inst* cannot find the client's name in a hosts database. If NIS=no in the configuration file for this client, check the server's local */etc/hosts* file for the client's entry. If NIS=yes in the configuration file for this client, check the NIS hosts database for the client and verify that the diskless server is an NIS client.
<*clientname*> `is not a diskless client for CLASS classname`	You are trying to remove a client from the wrong diskless server or *client_inst* cannot determine the client-server relationship. If NIS=no in the configuration file for this client, check the server's local */etc/bootparams* file for correct entries. If NIS=yes in the configuration file for this client, check the NIS *bootparams* database and verify that the diskless server is an NIS client.

Removing a Diskless Class Manually

When you cannot use *share_inst* or *client_inst* to remove diskless software, use this procedure to remove a diskless tree manually. Do this procedure as the superuser on the server.

1. Remove the share tree.

   This command sequence changes to the root directory for share trees and removes the share tree *IRIX_51*, using the recursive option of the *rm* commands:

   ```
   # cd /diskless/share
   # rm -r IRIX_51
   ```

2. Remove the client trees.

   This command sequence changes to the root directory for client trees and removes the client tree for *starlite*. Repeat the *rm -r* command for each client you need to remove:

   ```
   # cd /diskless/client
   # rm -r starlite
   ```

3. Remove the swap trees in the class.

   This command sequence changes to the root directory for swap trees and removes the swap tree for *starlite*. Repeat the *rm -r* command for each client you need to remove.

   ```
   # cd /diskless/swap
   # rm -r starlite
   ```

4. Clean the */etc/exports* file on the server.

   The share and client tree entries are listed in the server's */etc/exports* file. Delete them individually to prevent unintentional deletion of other information, as shown in these examples:

   ```
   # export -u /diskless/client/starlite
   # export -u /diskless/swap/starlite
   # export -u /diskless/share/IRIX_51
   ```

5. Clean unwanted entries in the */etc/bootparams* file on the server.

   If you want to retain other share trees, be very careful. You should delete only the individual entries for an unwanted client or share tree. An */etc/bootparams* entry looks like the following example.

   ```
   dayglo.bldg9.dude.com root=babylon:/d1/client/dayglo \
       sbin=babylon:/d1/share/5_1/sbin swap=babylon:/d1/swap/daglo\
   5_1.a109 root=babylon:/diskless/sh109/5_1.a109 sbin=babylon: swap=babylon:
   ```

Debugging the Boot Process

Occasionally, a client does not boot after a diskless software installation or upgrade. A number of factors can contribute to this problem. To diagnose booting problems, it is helpful to understand the boot process in some detail. Each phase of the boot process is described below.

Phase 1: The Boot Request

In phase 1, the diskless client initiates a *bootp* request. This request can be in the form of either a broadcast, going to all servers on the same network, or a specific request to a known diskless server. The transport mechanism is the User Datagram Protocol (UDP/IP).

Once the *bootp* request reaches the diskless server, the super server *inetd(1M)* starts the *bootp* server. *bootp* is responsible for resolving the client's identity by examining its configuration file, */var/bootptab*, the */etc/ethers* file, and the */etc/hosts* file, in that order. The *bootp* server is also responsible for resolving the boot file name, which is specified in the client's EPROM variable, *bootfile*.

Phase 2: The Boot File Download

When the *bootp* server has resolved the client identity and boot file location, it calls *tftpd* (the Trivial File Transport Protocol daemon) to download the boot file, */var/boot/client_name/unix*. The client receives the kernel and loads it into memory.

Phase 3: The Server-Client Setup

Once the client has loaded the kernel into memory, it initiates a broadcast *rpc.bootparams* call to the server. The *inetd* daemon on the diskless server starts up the *bootparamd* server. The *bootparamd* server provides the information to the diskless client necessary for booting; primarily, the location of its root, share, and swap trees. This information is contained in the *bootparams* database file, */etc/bootparams*(4), either locally or in the NIS databases. The server-client relationship is then sent back to the client.

Phase 4: Client Request for Software

After the server-client relationship is confirmed by means of the *bootparams* database, the client issues a mount request to the server for its *root*, *sbin*, and *swap* filesystems. The *inetd* server then starts the *rpc.mountd* server to answer the request. The *rpc.mountd* server is responsible for verifying access to a filesystem by clients and users; it verifies access by reading the */etc/xtab*(4) file. If the client has sufficient permission to mount the requested filesystems, the mount and boot processes are completed.

Figure 7-1 explains the events in the boot process.

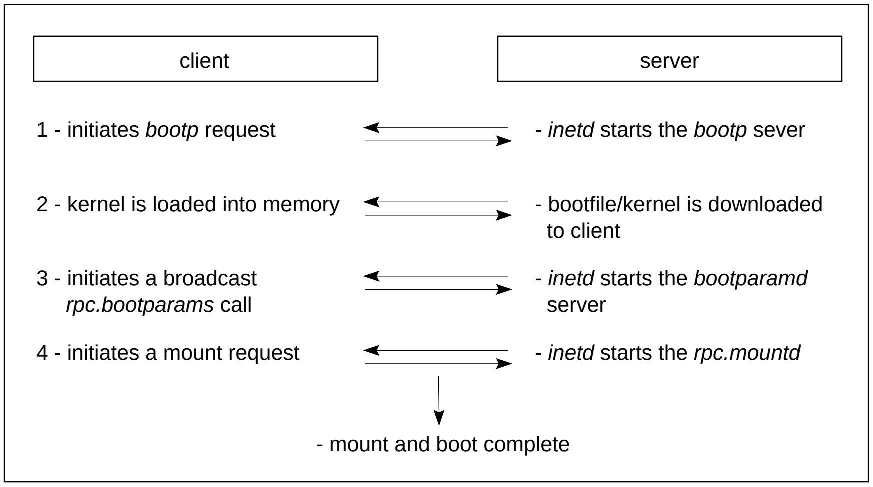
client
server
1 - initiates *bootp* request
- *inetd* starts the *bootp* sever
2 - kernel is loaded into memory
- bootfile/kernel is downloaded to client
3 - initiates a broadcast *rpc.bootparams* call
- *inetd* starts the *bootparamd* server
4 - initiates a mount request
- *inetd* starts the *rpc.mountd*
- mount and boot complete

Figure 7-1 The Diskless Boot Process

Handling Booting Error Messages

To see error messages that occur during the client’s boot process, the NVRAM *verbose* variable must be on. To set *verbose* to *on*, use this command:

```
>> setenv verbose on
```

Table 7-2 describes common error messages that you can see on the client during the boot process and suggests how to correct them.

Table 7-2 Boot Process Troubleshooting Hints

Client error messages:	**Check here**
[ec0, enp0, et0, ipg0, fxp0]: transmit: no carrier *no server for server: /var/boot/client/unix* *Unable to continue: press <enter> to return to the menu:*	Check the network cable to be sure that it is securely attached.
no server for server: /var/boot/client/unix *Unable to continue: press <enter> to return to the menu:*	On the client, check the NVRAM *netaddr* variable to ensure that it has the correct internet address. Ensure that the client and server are on the same physical and logical (internet) network. On the server, check the */etc/inetd.conf* (1M) file to ensure that the *bootp*(1M) server is supported.
File /var/boot/client/unix not found on server servername *Unable to continue: press <enter> to return to the menu:*	On the client, the NVRAM bootfile variable might have the wrong path or bootfile name. On the server, check to ensure bootfile specified in bootfile variable on client actually exists.
Starting up the system ... *Copyright message...* *PANIC: KERNEL FAULT......*	On the client, the *diskless* variable may not be enabled. The kernel built for this client workstation might not be appropriate for its model type. Check the architecture variables in *client.dat* and *server.dat* to be sure the correct kernel modules have been specified. On the server, the *swap* filesystem may not be exported correctly to the client; check the server's */etc/exports*(4) file.
Starting up the system... *Unable to continue: press <enter> to return to the menu:*	On the server, check the */etc/inetd.conf* file to ensure that the *tftpd*(1M) server is supported and configured correctly. The default *tftpd* entry runs in secure mode and supports the diskless environment. The default entry is `tftpd -s /var/local/boot`
kernel mount failed, check server, bootparams *or press reset button!!!* *Get_bootp failed*	On the server, check the */etc/inetd.conf* file to ensure that *bootparamd*(1M) is supported.

Table 7-2 (continued) Boot Process Troubleshooting Hints

Client error messages:	Check here
GET_BOOTP: WHOAMI fail, addr=0x *Get_bootp failed* *Kernel mount failed, check server, bootparams or press reset button!!!*	On the server, check to be sure that the *domainname* (1) command returns a value (even if you are not running NIS, the server must be assigned a domain name). On the server or NIS master, check that the */etc/bootparams* file contains valid entries for the client. If the client was created with NIS=yes in *client.dat*, the problem is with the NIS bootparams database. If the client was created with NIS=no in *client.dat*, the problem is with the server's local */etc/bootparams* file.
sv1: missing *Portmapper not responding; still trying*	On the server, check the *HOSTNAME* setting in *share.dat* to be sure that it specifies the network interface to the diskless LAN. If the diskless LAN is on a secondary network interface, *HOSTNAME* should not be set to "'*hostname*'" unless */etc/sys_id* contains the name of the diskless LAN interface.
Kernel mount failed, check server, bootparams or press reset button!!! *Diskless root mount failed:*	On the server, check the */etc/exports*(1M) file. The diskless tree filesystems (*root*, *swap*, and *share*) may not be accessible to clients.
nfs_rmount: SHARE MOUNT FAILED.	On the server, check the */etc/exports*(1M) file. The share tree for this client may not be accessible.

Handling Performance Problems

Some system configurations, particularly client workstations containing eight or sixteen megabytes of memory, appear hung when they are swapping through the Ethernet at a very heavy volume. This condition is called a *swap storm*. When an application is loading large files, such as bit-mapped graphics files, a swap storm might cause the system to take as long as 5 minutes to complete swapping.

This problem has several solutions. Adding main memory decreases the severity of swap storms; however, it does not eliminate them entirely. Swap storms can occur on virtually any system if memory is sufficiently overloaded. The best remedy for a swap storm (or hung client) is to avoid the activity that seems to cause it.

Before you try to correct a swap storm, verify that a swap storm is in progress. You can check for a swap storm in the following ways:

1. Move the mouse and check for cursor movement.

   If the cursor moves, the system is not hung, although nothing else seems to happen. The graphics engine runs fairly independently of system software and is usually able to react if the system is still functioning.

2. Run *gr_osview* on systems that appear to be hung.

   If, while the system appears hung, *gr_osview* shows that swapping is occurring (check the yellow segment of the I/O bar), then a swap storm is likely occurring. However, there is no guarantee that *gr_osview* will show swapping activity at this point, since *gr_osview* may not have had time to update before the swap storm began.

   Note: If you are working at a non-graphics console, use *osview(1)* rather than *gr_osview.*

3. Wait a few minutes for the behavior to stop.

   Sometimes the waiting period takes as long as a half hour. To be certain about a swap storm, allow a full hour for the system to recover. If the symptoms are still occurring, the problem is probably caused by hardware. If the symptoms have disappeared after a full hour, the problem was probably caused by a swap storm.

Before You Call for Help

Silicon Graphics support organizations are interested in your problems and are eager to help. However, before you call, please use the recommendations in this chapter for solving your problems independently. If your problems persist and you find it necessary to call, please have this information ready:

- Make printed copies of the */etc/exports* and */etc/bootparams* files for the server (not the share tree).
- Make printed copies of all *share.dat* and *client.dat* files in the server's */var/boot* directory—one for each client class. Label them so that you can distinguish among them.
- Make a list of all of diskless clients that you have on the network, including workstation models, graphics board types, and host names.

Appendix A

Determining the Client Architecture

If a workstation model that you plan to use as a diskless client is not listed in Table 2-2 or in the *client.dat* file, you can use two other methods to determine client architecture: using the hardware inventory command, *hinv,* and using the workstation model number. This appendix explains how to use each method to make the appropriate entries in your working copies of *share.dat* and *client.dat*.

Note: The information listed in this appendix pertains to older workstation models and other models that are not supported in the IRIX 5.1 Diskless Workstation implementation.

This appendix contains the following subsections:

- "Using the Hardware Inventory Command"
- "Using the Workstation Model Number"

Note: If the EPROM in a client workstation does not support the *hinv* command and you do not know the workstation model number, you will need to add a local disk to the workstation and install the IRIX mini-root. Use the output of mini-root *hinv* and Table A-1 and Table A-2 to determine the entries for the working copy of *client.dat* for this workstation.

Using the Hardware Inventory Command

Table A-1 shows you how to use the output of the *hinv* command to set values for the *CPUARCH, CPUBOARD* and *MACH* variables in your working copy of *share.dat* and *client.dat*. Settings for these variables are based on the client's processor type, which usually appears in the first line of *hinv* output.

Note: Remember that the output from *hinv* is not always consistent with the values that you enter in the configuration file. These differences are due to software requirements.

Table A-1 Setting CPU Variables from *hinv*

If processor type is	CPUARCH	CPUBOARD/MACH
IP4	R3000	IP4
IP5	R3000	IP5
IP6	R3000	IP6
IP7	R3000	IP5
IP9	R3000	IP5
IP17	R4000	IP17
IP19	R4000	IP19

Table A-2 shows how to use the output of the *hinv* command to set values for the *GFXBOARD* and *SUBGR* variables in your working copy of *server.dat* and *client.dat*.

Table A-2 Setting Graphics Variables from *hinv*

If command output is	GFXBOARD	SUBGR
Graphics board: VGR2	ECLIPSE	ECLIPSE
Graphics board: GR1*	ECLIPSE	ECLIPSE
GT Graphics option installed	CLOVER2	<cpu>GT[a]
VGX Graphics option installed	STAPUFT	IP7GT
VGXT Graphics option installed	STAPUFT	SKYWR
Graphics board: LGI	LIGHT	LIGHT
No graphics value shown	SERVER	SERVER

[a] Replace <cpu> with the *CPUBOARD* number

Using the Workstation Model Number

Table A-3 and Table A-4 provide the information you need to determine the CPU and graphics subsystems for any Silicon Graphics machine using its model number. Table A-3 identifies CPU subsystems, based on a base model number. Table A-4 identifies graphics subsystems, based on a model extension code.

Table A-3 shows you how to use the base model number of a workstation to set the CPUBOARD and MACH variables in your working copy of *server.dat* and *client.dat*. For example, the base model number of the 4D/120GTX workstation is *120*; you would set the *CPUBOARD* and *MACH* variables to *IP5* for these clients.

Table A-3 Setting CPU Variables from Model Numbers

Base Model	CPUARCH	CPUBOARD/MACH
RPC/30/35	R3000	IP12
20/25	R3000	IP6
50/60T/70/80/85	R3000	IP4
120	R3000	IP 5
210/220/240/280	R3000	IP5
310/320/340/380	R3000	IP5
420/440/480	R3000	IP5
Crimson	R4000	IP17
Onyx	R4000	IP19
Challenge	R4000	IP19
Challenge M	R4000	IP22

Table A-4 shows you how to use the model extension code to set the *GFXBOARD* and *SUBGR* variables in your working copy of *server.dat* and *client.dat*. For example, the model extension number on the 4D/120GTX workstation is *GTX*; you would set the *GFXBOARD* variable to *CLOVER2* and *SUBGR* variable to *IP5GT* for these clients.

Table A-4 Setting Graphics Variables from Model Numbers

Model Extension Codes: models end with...	GFXBOARD
letter S	"SERVER -mSUBGR=SERVER"
letters GT, GTX, or GTXB	"CLOVER2 -mSUBGR=<cpu>GT"[a]
letters VGX or VGXB	"STAPUFT -mSUBGR=IP7GT"
letters IG2 or IG2B,VGXT	"STAPUFT -mSUBGR=SKYWR"
Crimson Starter Graphics	"LIGHT -mSUBGR=IP17"
Crimson XS, XS24, Elan	"EXPRESS -mSUBGR=IP17"
Crimson Reality Engine	"VENICE -mSUBGR=IP17"
210/220/240/280	"VENICE -mSUBGR=IP17"
310/320/340/380	"VENICE -mSUBGR=IP17"
420/440/480	"VENICE -mSUBGR=IP17"

[a] Replace <cpu> with the CPUBOARD variable that you derived from Table A-3.

Index

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-0855-030.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

  Technical Publications
  Silicon Graphics, Inc.
  2011 North Shoreline Boulevard, M/S 535
  Mountain View, California 94043-1389