

Diskless Workstation Administration Guide

Document Number 007-0855-070

CONTRIBUTORS

Written by George Eckel, Steven Levine, and Pam Sogard

Illustrations by Dany Galgani

Engineering contributions by Hercules Huggins, Bill Kawakami, and Bob Sanders

Other contributions by Michael Nishimoto and Joe Yetter

St Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

© Copyright 1998-1999, 2005, Silicon Graphics, Inc.-- All Rights reserved

This document contains proprietary and confidential information of Silicon Graphics, Inc. The contents of this document may not be disclosed to third parties, copied, or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED AND RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in the Rights in Data clause at FAR 52.227-14 and/or in similar or successor clauses in the FAR, or in the DOD, DOE or NASA FAR Supplements.

Unpublished rights reserved under the Copyright Laws of the United States.

Contractor/manufacturer is Silicon Graphics, Inc., 1500 Crittenden Lane, Mountain View, CA 94043.

Silicon Graphics, SGI, the SGI logo, IRIS, Indy, 02, Octane, Onyx, Origin, Tezro, and IRIX are registered trademarks and Fuel, Indigo2, and Indigo Elan are trademarks of Silicon Graphics, Inc. NFS is a trademark of Sun Microsystems, Inc. R4000, R5000, and R10000 are registered trademarks of MIPS Technologie, Inc., used under license by Silicon Graphics, Inc. UNIX is a registered trademark of X/Open Company, Ltd.

Diskless Workstation Administration Guide

Document Number 007-0855-070

Contents

	List of Figures	vii
	List of Tables	ix
	List of Examples	xi
	About This Guide	xiii
	Product Requirements	xiii
	Summary of Contents	xiv
	Audience for This Guide	xv
	Notation Conventions	xv
	Related Documentation	xv
	Product Support	xvi
	New Features	xvii
1.	Understanding Diskless Operation	1
	Setting Up User Accounts	1
	Diskless Client-Server Model	1
	PROM Resources	3
	IRIX and IRIX Utilities	3
	NFS Resources	3
	Local Disk Resources	3
	Diskless Tree	4
	Share Tree	5
	Client Tree	6
	Swap Tree	6
	Influence of Client Classes	6
2.	Preparing for Installation	7
	Obtaining a Software Distribution	8

- Checking a Server 8
- Checking a Client 11
- Assigning Internet Addresses 13
- Planning the Diskless Tree 14
 - Diskless Filesystems 14
 - Using Multiple Filesystems 15
 - Planning Share Trees 15
 - Planning Client Trees 17
 - Planning Swap Trees 19
- Working With Diskless Software 20
 - Using inst to Install a Diskless Tree 21
 - Automatic File Updates 21
- Overview of the Installation Procedure 22
- 3. Creating a Share Tree 25**
 - Share Tree Configuration File 26
 - Server Variables in the Share Tree Configuration File 27
 - Share Tree Variables in the Share Tree Configuration File 29
 - Client Hardware Variables in the Share Tree Configuration File 30
 - Creating a New Share Tree Configuration File 33
 - Editing a Share Tree Configuration File 36
 - Verifying the Share Tree Configuration File 36
 - Installing the Share Tree 37
 - share_inst Command 37
 - Share Tree Installation Procedure 38
- 4. Creating Client Trees 43**
 - Client Tree Configuration File 44
 - Client and Swap Tree Variables 46
 - Client Hardware Variables 48
 - Creating a New Client Tree Configuration File 52
 - Editing the Client Tree Configuration File 54
 - Verifying the Client Tree Configuration File 54

Installing Client and Swap Trees	55
The client_inst Command	55
Client and Swap Tree Installation Procedure	56
Cloning Client Trees	59
Booting a Client Workstation	60
5. Updating Software on Diskless Trees	63
Points to Keep in Mind	63
Updating a Share Tree	64
Updating Existing Client Trees	67
When New Clients Need Updated Trees	68
Software Options as Updates	68
Adding Options to a Share Tree	68
Adding Options to Client Trees	70
6. Maintaining Diskless Implementations	73
Removing a Client Tree	74
Removing a Share Tree	75
Changing the Size of the Swap Space	76
Setting Up a Local Disk	76
Getting Prepared	77
Adding a Local Filesystem	78
Adding a Local Swap Partition	79
Moving a Client to a Different Server	79
Preparing the Client	80
Backing Up the Old Client Tree	80
Creating a Client Tree on the New Server	81
Moving Private Files to the New Tree	81
Verifying Client Operation	82
Cleaning the Old Server	82

- Converting a Client to a Standalone Workstation 83
 - Installing the Hardware 83
 - Enabling Network Software 85
 - Moving Private Files to the New Host 86
 - Removing the Client Tree 86
- 7. **Troubleshooting Diskless Installations** 87
 - General Approach to Troubleshooting 87
 - Troubleshooting Checklist 88
 - Installation Error Messages 89
 - Removing a Diskless Class Manually 90
 - Debugging the Boot Process 91
 - Phase 1: The Boot Request 91
 - Phase 2: The Boot File Download 91
 - Phase 3: The Server-Client Setup 92
 - Phase 4: Client Request for Software 92
 - Handling Booting Error Messages 93
 - Before You Call for Help 94
- Index** 95

List of Figures

Figure 1-1	Elements of Diskless Service	2
Figure 1-2	Simple Diskless Tree	4
Figure 1-3	Diskless Tree with Multiple Share Trees	5
Figure 2-1	Level 1 and 2 Directories in the Diskless Trees	14
Figure 2-2	Typical Share Tree Contents	15
Figure 2-3	Typical Client Tree Organization and Contents	18
Figure 2-4	Typical Swap Tree Organization	20
Figure 2-5	Flow Diagram of Installation Tasks	23
Figure 3-1	Hardware Modules for Multiple Classes	30
Figure 4-1	Multiple Client Trees From One Configuration File	45
Figure 7-1	Diskless Boot Process	92

List of Tables

Table 2-1	Workstation Hardware Components	12
Table 2-2	Space Requirements for a Basic Share Tree	16
Table 6-1	IRIX Disk Partitions	78
Table 7-1	Installation Error Messages	89
Table 7-2	Boot Process Troubleshooting Hints	93

List of Examples

- Example 3-1** Server Specification Section of share.dat 27
- Example 3-2** Share Tree Specification Section of share.dat 29
- Example 3-3** Hardware Specification Section of share.dat 31
- Example 4-1** Client Tree Setup Section of client.dat 46
- Example 4-2** Architecture Tables in client.dat 48

About 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 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.
- A diskless server must be running an IRIX release that is the same as or later than the IRIX release used by any of its clients.

Summary of Contents

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

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 client and server systems 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 use the program that automatically creates a share tree. Use Chapter 3 when installing a new share tree. Do not use Chapter 3 to update an existing tree with a new software release (see Chapter 5).

Chapter 4, "Creating Client Trees"

Explains how to use the program that automatically creates a client tree. 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 (see Chapter 5).

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.

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 is helpful, but not required, for installing and maintaining Diskless 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 is required to prepare network software for the diskless implementation.

Notation Conventions

This document uses the standard UNIX convention when referring to entries in IRIX reference pages. 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 reference pages.

File and directory names and 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*:

IRIX Admin: Software Installation and Licensing

IRIX Admin: Disks and Filesystems

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.

New Features

This version of the manual incorporates updated information on configuration and hardware support. It now documents support for IP35 CPU boards and the Silicon Graphics Fuel and Tezro workstations.

Note that when creating a new share tree configuration file, as described in “Creating a New Share Tree Configuration File” on page 33, selecting yes when asked if you want to install a share tree for all client platforms will result in an Xserver error on Fuel and Tezro systems. To avoid this, select IP35 as the specified client platform and limit supported client platforms to no more than three platforms for a share tree that includes Fuel and or Tezro systems as target clients.

Understanding Diskless Operation

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

- “Setting Up User Accounts” on page 1
- “Diskless Client-Server Model” on page 1
- “Diskless Tree” on page 4
- “Influence of Client Classes” on page 6

Setting Up User Accounts

The general procedure for setting up user accounts is provided in *IRIX Admin: System Configuration and Operation*, in the section titled, “Setting Up a Home Directory for a New User.” Because the `/usr` directory is readonly mode on diskless systems, you should either place home directories in an alternate location or make the `/usr/people` directory in the share tree a symlink to another directory (`/home` is a conventional location).

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 relationship between the *server* (the computer supplying the resources) and the *client* (the computer using the resources) 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 a diskless configuration and the process by which clients obtain boot files from a server.

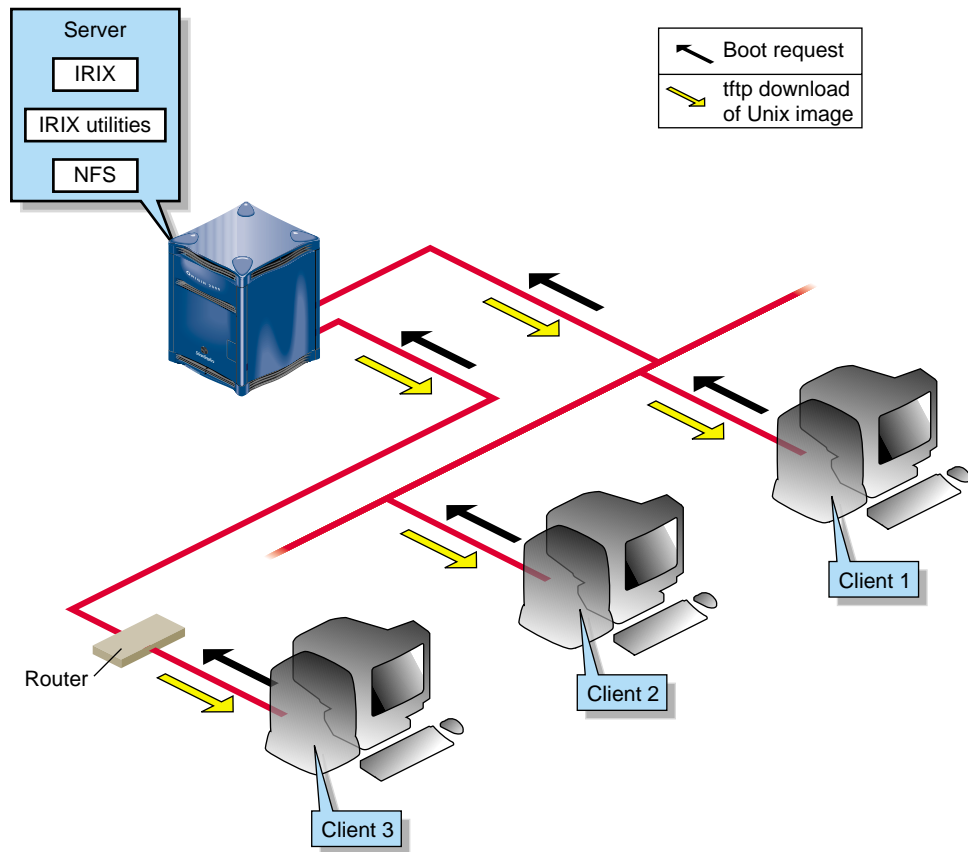


Figure 1-1 Elements of Diskless Service

PROM Resources

The PROM on a diskless client must include the Internet Bootstrap Protocol, *bootp*, the network protocol that initiates remote booting. During the client's power-up sequence, *bootp* sends a request for a boot file (the *unix* image) to the diskless server. The server downloads the client's boot file using the Trivial File Transfer Protocol, *tftp*. (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. The swap space that is 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 the NFS 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.

Diskless Tree

The software on a server that is dedicated to supporting its diskless clients is referred to as the *diskless tree*. Ordinarily, 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 trees*, *client trees*, and *swap trees*. The overall size and configuration of a diskless tree depends on the number of clients that 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.

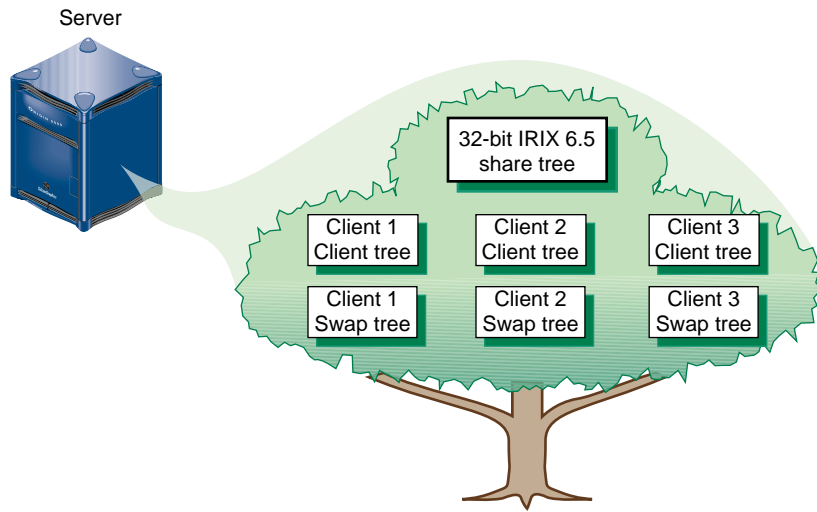


Figure 1-2 Simple Diskless Tree

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 when it includes more than one client.

A diskless tree contains one share tree for each release of IRIX that it supports. Two share trees are required to run a 32-bit and a 64-bit version of the same operating system (an IRIX 6.5 share tree for 32-bit clients and a separate IRIX 6.5 for 64-bit clients, for example). 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.

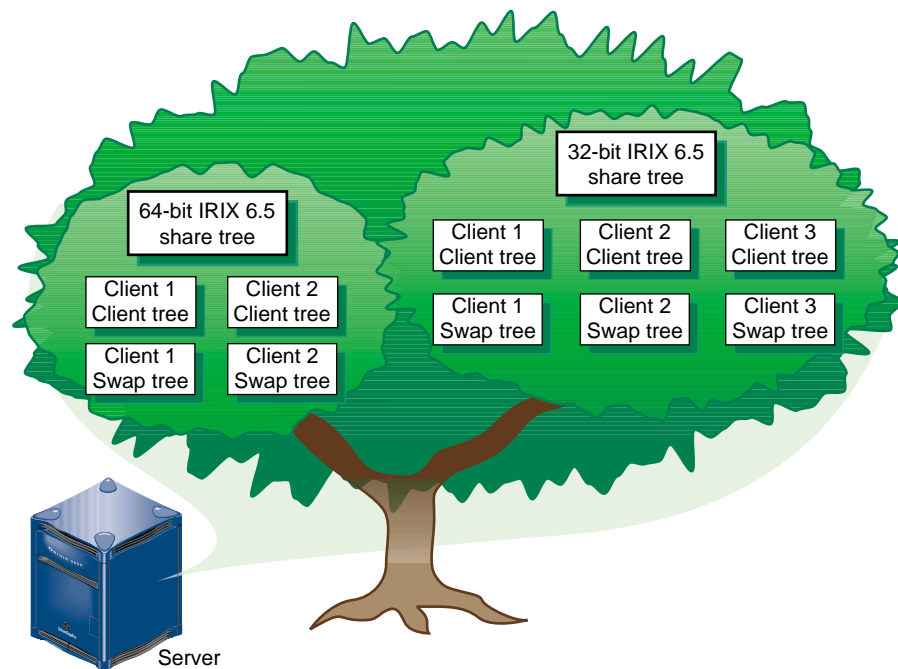


Figure 1-3 Diskless Tree with Multiple Share Trees

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.

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 (details are explained in “Adding a Local Swap Partition” on page 79).

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 O2™ or Octane™.

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 (details are explained in “Client Hardware Variables in the Share Tree Configuration File” on page 30).

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 evaluate and change to get ready for the diskless service, and how to organize the server filesystem that will contain the diskless software. It also provides an overview of diskless software and the installation procedure.

This chapter contains the following sections:

- “Obtaining a Software Distribution” on page 8
- “Checking a Server” on page 8
- “Checking a Client” on page 11
- “Assigning Internet Addresses” on page 13
- “Planning the Diskless Tree” on page 14
- “Working With Diskless Software” on page 20
- “Overview of the Installation Procedure” on page 22

Obtaining a Software Distribution

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 with read-only memory (CD-ROM). On some networks, a copy of this software is available from a central distribution source on the network. You can use either source to install diskless software, as long as you use an official software release.

At minimum, the IRIX 6.5 source must include the Silicon Graphics Execution Only Environment product (*ee*) and its supporting software, and the NFS product option, *nfs*. A group of subsystems from each of these products is required to install diskless software (see Table 2-2 for details).

Checking a Server

Use the questions that follow to evaluate 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 1.5 gigabyte (GB) of disk space is required for a diskless tree with these characteristics:

- One share tree of IRIX 6.5 with a minimum software configuration (see Table 2-2)
- Five client trees with a minimal software configuration
- Five swap trees, each containing 128 megabytes (MB) of swap space

3. Is the server running the necessary software?

The IRIX version implemented on a server must be the same as or more recent than the IRIX version of any share tree that it supports. In addition, a server running 32-bit IRIX cannot support share trees of 64-bit IRIX. See the release notes shipped with your software for more specific compatibility information.

To determine the version of IRIX running on a server, use the `uname` 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
6.5
```

You should also verify that the host acting as the diskless server is running `nfs.sw.diskless_server`, the software subsystem that supports the diskless server function. To verify, enter the `versions` command and look for the diskless server subsystem in the command bitoutput, as shown in this example:

```
% versions nfs
I  nfs.sw.dskless_server  08/18/97  Diskless Server Support
```

4. Is the TFTP environment set up correctly?

Client boot files are transferred from the server by means of the trivial file transfer protocol (TFTP). The server must contain a user account that owns TFTP (see the `tftp` reference page). By default, the owner of TFTP is `guest`. If you want a user other than `guest` to own TFTP (for security reasons, for example), you must modify the file `/etc/inetd.conf` to specify a different owner (see the `inetd` reference page). 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`).

You must add `/var/boot` to the end of the `tftp` entry in the `/etc/inetd.conf` file; this is because in order to boot a diskless client the path defined as `/var/boot` needs to be defined for TFTP to allow the client to find its kernel.

Additionally, the second `tftp` line containing `udp6` should be commented out.

5. Is an NIS domain name assigned to the server?

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` command:

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

If this command returns 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, enter this command as the superuser:

```
# domainname fakedomain
```

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

```
# echo fakedomain > /var/yp/ypdomain
```

The system looks for the file `ypdomain` that contains the domain name.

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 prevents the installation scripts from exporting diskless directories correctly and can also compromise security on the exported files. For example, if the diskless tree is installed in the directory `/usr/diskless`, the `/usr` directory should not be exported. To check for export conflicts, examine the server's `/etc/exports` file. If the parent directory of the diskless tree is listed, adjust the NFS configuration before implementing the diskless server.

7. Is the notation in the host database correct?

Client hostnames must be listed in the server's `/etc/hosts` in 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. Can BOOTP requests from clients reach the server?

If the server is on a different network from any of its clients, BOOTP forwarding must be enabled on the routers between the networks. See question 1 in the "Checking a Client" on page 11 for details.

In addition, it may be desirable to edit */etc/inetd.conf* to uncomment the simpler *bootp* line and comment out the *bootp* line containing the DHCP options.

9. Is bootparams enabled on the server?

If the *bootparams* argument commented out in the *inetd.conf* file, it prevents the server from responding to client requests for boot information. Remove the comment character (#) to set bootparams on.

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?

Client systems initially request critical information from the server using the Internet Bootstrap Protocol (BOOTP). If a client and its server are located on different networks, the routers connecting the networks must be configured to perform BOOTP forwarding (see the *bootp(1M)* reference page for details). If you do not have responsibility for configuring network routers, contact your network administrator about this router configuration requirement.

2. What hardware components does the client contain?

When you prepare the share tree configuration file, it helps to know the model names of your diskless workstations and the designations for the hardware components that the workstations contain. You can use the *hinv* command at client workstations to get component information. Make a list of these components to help you during the procedures in Chapter 3.

Table 2-1 shows the components of Silicon Graphics workstation models.

Table 2-1 Workstation Hardware Components

Model Name	CPU Processor	CPU Board	Graphics Subsystem	Graphics Subgroup	Video Component
Indigo	R4000®	IP20	LIGHT	LG1MC	None
Indigo Elan	R4000	IP20	EXPRESS	EXPRESS	None
Indigo2 XZ	R4000	IP22	EXPRESS	EXPRESS	None
Indigo2 Extreme	R4000	IP22	NEWPRESS	NEWPRESS	None
Indigo2 XL	R4000	IP22	NEWPORT	NG1	None
Indigo 2 IMPACT	R4000	IP22	MGRAS	MGRAS	None
Indy	R4000	IP22	NEWPORT	NG1	VINO
Indy R5000	R5000	IP22	NEPORT	NEWTON	VINO
Power Indigo2	R10000	IP28	EXPRESS	I2_64	None
Power Indigo2 IMPACT	R10000	IP28	MGRAS	I2_64	None
Octane	R10000	IP30	MGRAS	RACER	None
O2	R5000	IP32	CRIME	CRM	VICE
O2	R10000	IP32	CRIME	CRM	VICE
Fuel	R10000	IP35	ODYSSEY	NULL	NULL
Tezro	R10000	IP35	ODYSSEY	NULL	NULL
Onyx350	R10000	IP35	KONA	NULL	NULL
Origin300	R10000	IP35	SERVER	SERVER	NULL
Origin350	R10000	IP35	SERVER	SERVER	NULL
Origin3000	R10000	IP35	SERVER	SERVER	NULL
Onyx4	R10000	IP35	VOYAGER	VOYAGER_1_0	NULL

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

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 services.

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 Admin: Networking and Mail* 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.

Diskless Filesystems

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 server's */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.

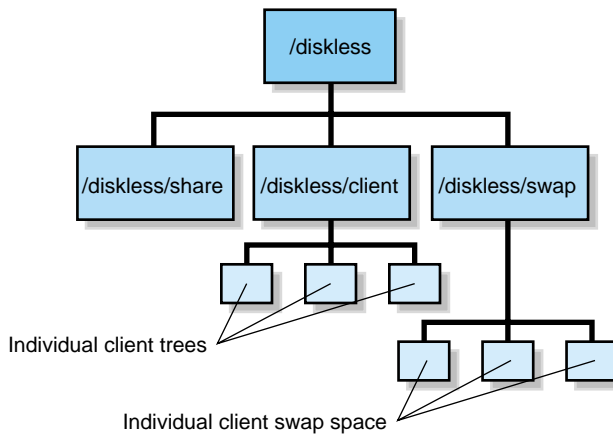


Figure 2-1 Level 1 and 2 Directories in the Diskless Trees

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 specify the complete pathnames for the share tree, client tree, and swap location when you generate diskless configuration files (see *DISKLESS* and *SHARE* descriptions in “Share Tree Variables in the Share Tree Configuration File” on page 29 and see *CLROOT* and *SWAP* descriptions in “Client and Swap Tree Variables” on page 46).

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.

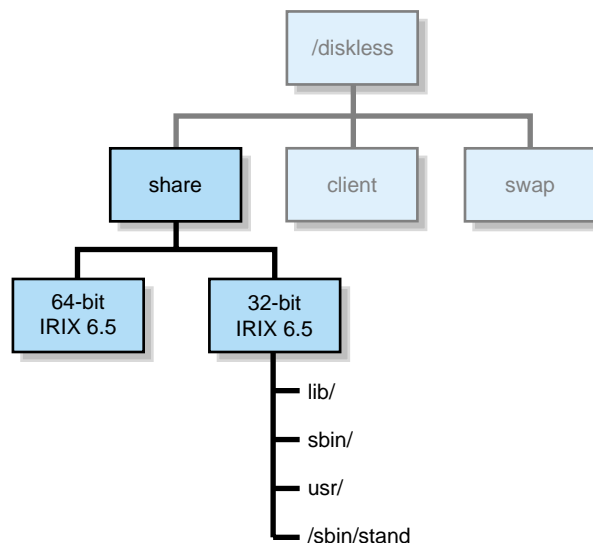


Figure 2-2 Typical Share Tree Contents

A share tree of IRIX 6.5 that supports clients with a minimum software configuration and the IRIX desktop requires approximate 700 megabytes of disk space. However, 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; instead, mount the applications from other NFS servers where they are already installed.

Note: If you plan to mount NFS directories from a server other than the diskless server, you must edit the automatically generated copy of */etc/fstab* (describe in “Automatic File Updates” on page 21) to contain these entries.

Table 2-2 provides a list of the required subsystems for an IRIX 6.5 share tree and the size and description of each subsystem, assuming a 64-bit client such as Silicon Graphics Tezro or Silicon Graphics Fuel.

The base minimal share tree size is 695,932 KB.

Table 2-2 Space Requirements for a Basic Share Tree

Subsystem Name	Size (KB)	Description
4Dwm	816	Desktop Window Manager Man Pages
desktop_base	10,728	Desktop Base Software
desktop_eoe	14,296	Desktop Environment
desktop_tools	1,716	Desktop Tools
eoe	307,804	IRIX Execution Environment
nfs	8,340	NFS Support
x_eoe	119,388	X11 Execution Environment
Viewkit_eoe	10,804	Viewkit Execution Environment
c++_eoe	5,900	Standard Execution Environment (C++)
compiler_eoe	47,864	IRIX Standard Execution Environment (Base Compiler)
dmedia_eoe	125,044	Digital Media Execution Environment
ifl_eoe	17,940	Image Format Execution Environment
il_eoe	44,628	ImageVision Library Execution Environment
media_warehouse	2,032	Desktop MediaWarehouse

Table 2-2 (continued) Space Requirements for a Basic Share Tree

Subsystem Name	Size (KB)	Description
motif-ee	30,960	IndigoMagic Execution Software (Motif)
tooltalk_ee	9,492	ToolTalk Execution Environment
websupport_ee	6,036	Websupport
webviewer	12,588	WebViewer Library Execution Environment
field_diags_sysco	39,484	Standalone System Controller Environment
insight	1,104	Online Documentation Software
insight_base	8,532	Online Documentation Base Software
inventor_ee	42,444	Inventor Execution Only Environment
kerberos	9,404	MIT Kerberos
openssh	9,016	OpenSSH Server and Client
openssl	22,620	OpenSSL Libraries
accessx	396	Access for Movement-Impaired Users
mam	3,112	Motif Accessibility Module
netscape	4,000	Netscape Communicator Client

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.

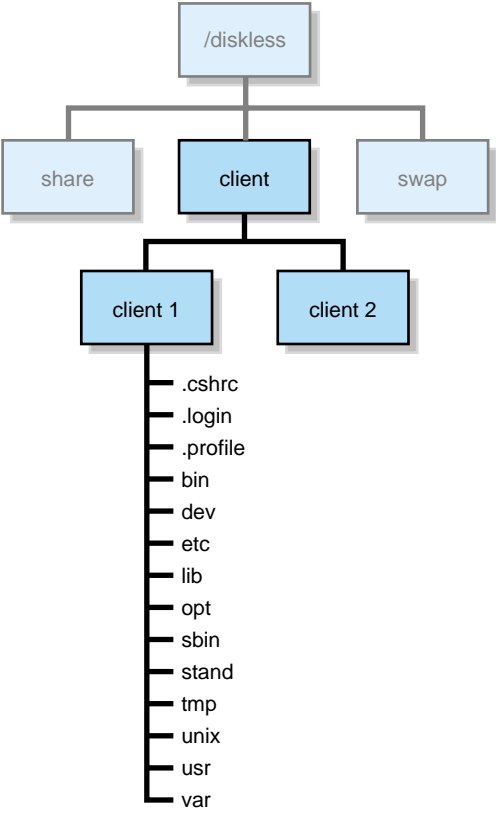


Figure 2-3 Typical Client Tree Organization and Contents

A newly installed client tree requires approximately 58 megabytes of disk space. However, because the client tree allows read and 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.

Note: If you plan to mount NFS directories from a server other than the diskless server, you must edit the automatically generated copy of */etc/fstab* (describe in “Automatic File Updates” on page 21) to contain these entries.

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 64 megabytes. But swap space size is a configurable variable that you can set during a client tree installation 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 gui applications for email, news, and web surfing are the principal applications, consider increasing swap space to 128 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.

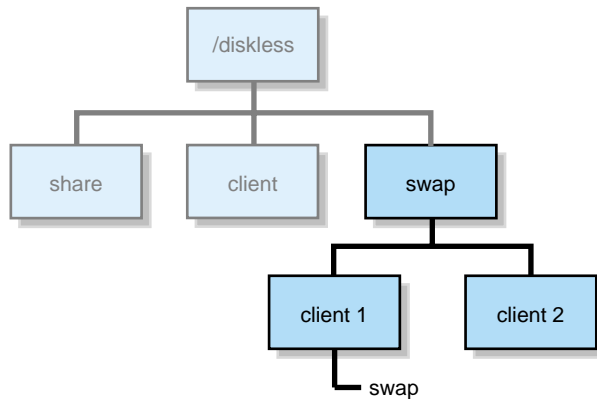


Figure 2-4 Typical Swap Tree Organization

Working With Diskless Software

Diskless software consists of installation utilities and configuration files that set up the share tree, client trees, and swap trees for a diskless implementation:

/var/boot/share_setup

The program that you use to create a new share tree configuration file.

/var/boot/share_inst

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

/var/boot/share.dat

The default 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. A working copy of this file is created automatically when you run *share_setup*.

/var/boot/client_setup

The program that you use to create a new client tree configuration file.

/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 default 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 must contain a working copy of *client.dat* for each class of client workstations (see “Influence of Client Classes” on page 6). A working copy of this file is created automatically when you run *client_setup*.

/var/boot/clone_client

The program that reproduces client trees after one client tree is created. This program creates one or more cloned client trees from an existing client tree or from a share tree. A cloned tree supports clients of the same class as that supported by the original tree. For example, if an existing O2 client tree is cloned, the cloned tree cannot be used for Indy clients.

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. You can use either user interface to *inst*: Software Manager, the graphical interface; or *Inst*, the command line interface. Once *inst* is invoked from either interface, 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. The *bootparams* file specifies the names of diskless clients and the information needed to boot them to.

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. Run *share_setup* to create a share tree configuration file (*share.dat*).
2. Run *share_inst* to install the share tree.
3. Run *client_setup* to create a client tree configuration file (*client.dat*).
4. Run *clone_client* to reproduce the client and swap trees for additional clients.
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 (see “Influence of Client Classes” on page 6). If you are installing multiple IRIX releases, you need to repeat the entire procedure for each IRIX version that you install (remember that 32-bit and 64-bit versions of the same IRIX release are considered two separate operating systems in a diskless implementation).

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, because 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.

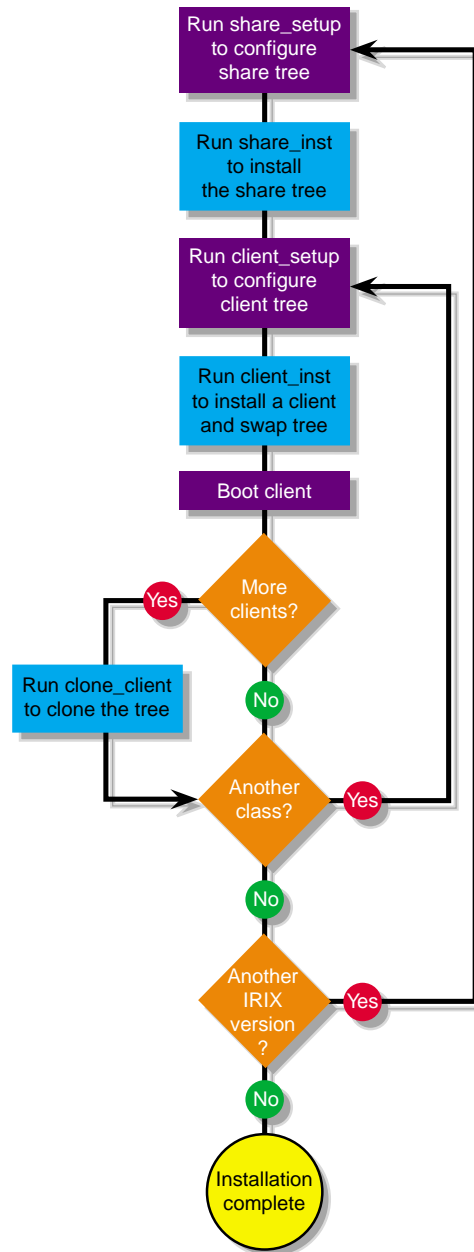


Figure 2-5 Flow Diagram of Installation Tasks

Creating a Share Tree

To create a new share tree, you run the `/var/boot/share_setup` program to create a share tree configuration file that specifies the requirements of your share tree. To modify a share tree configuration file that already exists, use a text editor. When the configuration file is finished, you run the share tree installation script, `/var/boot/share_inst`, which installs the share tree software based on the specifications that you provided in the configuration file. This process is repeated for each share tree that you plan to run (see Figure 2-5 for an illustration of this process).

This chapter provides the information and procedures that are needed to configure and install a share tree. It contains the following sections:

- “Share Tree Configuration File” on page 26
- “Creating a New Share Tree Configuration File” on page 33
- “Editing a Share Tree Configuration File” on page 36
- “Verifying the Share Tree Configuration File” on page 36
- “Installing the Share Tree” on page 37
- “share_inst Command” on page 37

Note: To update an existing share tree with a newer software version, use the information in Chapter 5, “Updating Software on Diskless Trees.”

Share Tree Configuration File

The share tree configuration file contains a list of variables that specify the diskless server and its clients. Specifically, these variables identify the host that will act as the diskless server, the name and location of the share tree, and the hardware components for each client class that the tree supports (see “Influence of Client Classes” on page 6 for background information).

Note: Your server should already have a default copy of a share tree configuration file in */var/boot/share.dat* as a result of installing Diskless software.

To create a new share tree configuration file, you can run the */var/boot/share_setup* program, or you can create a working copy of the */var/boot/share.dat* file that was stored on the server during the Diskless software installation process. If you plan to modify an existing share tree configuration file, you must use a text editor to make your changes.

Share tree configuration files are stored in the */var/boot* directory. This directory contains a separate configuration file for each share tree that it supports. For example, if a server supports three share trees, then the */var/boot* directory contains three separate share tree configuration files.

By convention, a share tree configuration file is named for the release of IRIX that the tree contains; a *.dat* extension is appended to its name. For example, this file contains configuration information for a share tree of IRIX 6.5 for 32-bit systems:

```
/var/boot/IRIX_65-32.dat
```

Note: Configuration information for share trees supporting 32-bit and 64-bit versions of the same IRIX release must be stored in separate share tree configuration files.

Server Variables in the Share Tree Configuration File

Server variables in a share tree configuration file specify the host that will act as the diskless server and whether NIS will provide any client information. 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 name lookup
#                 during installation. Please note that after
#                 yp_boot_param runs, it takes time for the information
#                 to propagate back to the diskless server from the
#                 NIS master. Also, this option is only fully functional
#                 for SGI NIS masters.
#                 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 hostname 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 hostname *calvin*, and an Ethernet interface to the diskless LAN with the hostname *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 hostname and address from the NIS database. The default setting is *no*.

Set *SERVER_NIS* to *yes* if you want *share_inst* to obtain client names, addresses, and filesystem locations from the NIS hosts. When you set *SERVER_NIS* to *yes*, *share_inst* makes an entry for this client in the bootparams map on the NIS master server. This assumes that the master server is a Silicon Graphics system.

Note: When you use an NIS server, expect some delay before client entries are 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 enter filesystem locations for this client in the local server's *bootparams* file.

Note: 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.

Share Tree Variables in the Share Tree Configuration File

Share tree variables in a share tree configuration file identify the 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 to *\$DISKLESS* the value that you specified for *DISKLESS*, the previous variable. This setting assigns to *\$RELEASE* the name of the share tree, which you supply as an argument to the *share_inst* command when installing the share tree (see "Share Tree Installation Procedure" on page 38).

Client Hardware Variables in the Share Tree Configuration File

Client hardware variables in a share tree configuration file specify the architectural components of the client models that the tree supports. These entries determine the hardware-specific files that are included in the share tree. 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.

By default, *share_setup* assumes that a share tree supports all client classes listed in Table 2-1. However, to reduce the amount of disk space that the share tree requires, include only the client classes that will use the tree.

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

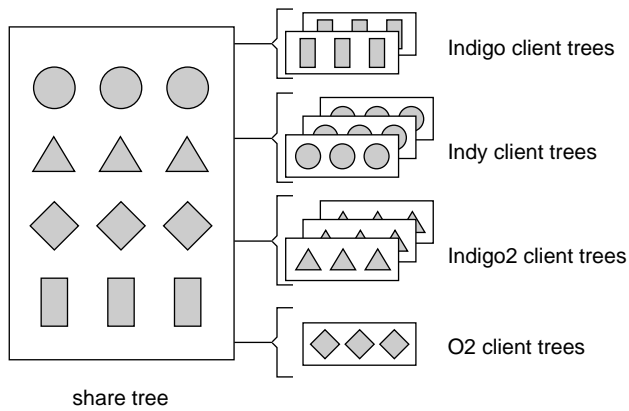


Figure 3-1 Hardware Modules for Multiple Classes

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.
#
# It is no longer possible to install a share tree with all possible
# architectures. The MODE variable must be set to either
# (32bit or 64bit). This is due to 32bit kernels or 64bit
# kernels running on the client. So a share tree can contain
# only 32bit kernels or 64bit kernels, but not both.
#
# To configure a share tree to install 32bit (or 64bit) kernels
# in the /usr tree, set the MODE variable to 32bit or 64bit. If a
# subset of client classes are to be installed, set all 5 variables
# for each class. The example below shows settings for an R4000
# Indigo with LIGHT graphics. (The MACH variable is obsolete.)

#CPUBOARD="IP20"
#CPUARCH=R4000
#GFXBOARD="LIGHT -mSUBGR=LG1MC"
#VIDEO=" "
#MODE=" "
#
# 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. The example below
# contains an R5k O2 with CRIME graphics and an R4k Indy with
# NEWPORT/NG1 graphics. (The MACH variable is obsolete.)
#
#CPUBOARD="IP20 -mCPUBOARD=IP12"
#CPUARCH="R4000 -mCPUARCH=R3000"
#MACH="IP20 -mIP12"
#GFXBOARD="LIGHT -mSUBGR=LG1MC -mGFXBOARD=EXPRESS -mSUBGR=EXPRESS"
#VIDEO=" "
```

Note: Use Table 2-1, 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 the share tree will support.

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*:

IP20
IP22
IP28
IP30
IP32
IP35

Set *CPUBOARD* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings to this variable (see Table 2-1 for details). Check the samples in Example 3-3 (or the *client.dat* file) for the correct format of your settings.

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

R4000
R5000
R10000

Set *CPUARCH* to the values needed by your clients. If your clients include different workstation models, you might need to assign multiple settings (see Table 2-1 for details). 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*:

LIGHT -mSUBGR= LG1MC
EXPRESS -mSUBGR= EXPRESS
NEWPRESS -mSUBGR= NEWPRESS
NEWPORT -mSUBGR= NG1
EXPRESS -mSUBGR= I2_64
MGRAS -mSUBGR= I2_64
MGRAS -mSUBGR= RACER
CRIME -mSUBGR= CRM
ODYSSEY -mSUBGR= IP35
SERVER -mSUBGR= SERVER
KONA -mSUBGR= IP35
VOYAGER -mSUBGR= VOYAGER_1_0

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.

<i>VIDEO</i>	Specifies the video chip. The default setting is null, which indicates a system with no on-board video hardware. You can assign these values to <i>VIDEO</i> :
	VINO
	VICE
<i>MODE</i>	Specifies the version of an IRIX release. You can assign these values to <i>MODE</i> :
	32bit
	64bit

Creating a New Share Tree Configuration File

Use the */var/boot/share_setup* program to create a separate configuration file for each share tree in your diskless implementation. Remember that a 32-bit and a 64-bit version of a particular IRIX release (such as IRIX 6.5) require separate share trees and therefore separate share tree configuration files.

Note: If you prefer to use an editor to create this file, or if you need to modify an existing share tree configuration file, use the information in “Editing a Share Tree Configuration File” on page 36 instead of this procedure.

This procedure explains how to create a new share tree configuration file with *share_setup*.

1. Create the root directory for the diskless tree.

See “Diskless Filesystems” on page 14.

2. Start *share_setup*.

```
# cd /var/boot
# ./share_setup
```

After your entry, you see this prompt:

```
This program will help you create a share tree configuration file
for a single IRIX version. Create a separate share tree
configuration file for a 32-bit and a 64-bit version of the same
IRIX release.
```

Retrieving system information...

Hostname is *hostname*

If you wish to use a different interface name for this host, enter it now (press return if hostname is ok):

3. Press Enter to accept the default interface name or enter a different interface name.

The *share_setup* program retrieves the name of the primary interface on the host where it is running. If clients are attached to a different interface, enter the other interface name at this prompt. Otherwise, press the Enter key to accept the hostname that *share_setup* retrieved. After you enter the interface name, this prompt is displayed:

Do you want client boot information stored on *hostname* (y/n)?

4. Specify whether client boot information should be stored on *hostname* or on an NIS host.

If client information is stored locally, answer yes to the previous prompt. This prompt is displayed after your entry;

Enter the root directory of the diskless tree (e.g. /diskless):

Client name and address information and information about client filesystems can also be stored on an NIS host in its */etc/bootparams* file. If this information is stored on an NIS host, answer no to the previous question.

This prompt is displayed after your entry (the *updaters* file allows users on the *hostname* system to write changes to the NIS database):

Make sure *hostname* is listed in the server's */etc/updaters* file

Enter the root directory of the diskless tree (e.g. /diskless):

5. Enter the name of the root directory for the diskless tree.

See "Diskless Filesystems" on page 14 if you need help. After your entry, this prompt is displayed:

Enter the name of the share tree (e.g. IRIX65):

6. Enter the directory name of the share tree.

Enter only the directory name for the share tree; do not include a pathname in your entry. The *share_setup* program will construct the pathname for you. After your entry, *share_setup* posts the complete pathname of the share tree (*share_dir* in the example below) and prompts for the name of the share tree configuration file:

Share tree will be installed at /diskless/share/*share_dir*

Enter the name of the share tree configuration file (no extension please):

7. Enter the filename for the share tree configuration file.

After your entry, *share_setup* posts the complete pathname of the share tree configuration file (*/var/boot/myfile.dat* in the example below) and prompts you to press Enter to continue:

```
Share tree configuration file will be written at
/var/boot/myfile.dat
```

```
Press [Enter] to continue
```

8. Press Enter to begin specifying the hardware components of your clients.

After your entry, this prompt is displayed:

```
***CLIENT ARCHITECTURE INFORMATION***
```

```
*** PLEASE READ CAREFULLY ***
```

If you are using a 32 bit machine as your diskless server, you can only install share trees for 32 bit platforms. If you have a 64 bit server, you can install either a 32 or a 64 bit share tree.

Do you want to install a share tree for all client platforms (y or n):

9. Answer yes or no to this question.

If you answer yes, *share_setup* creates a configuration file that contains all the software needed to support all classes of Silicon Graphics clients (see "Client Hardware Variables in the Share Tree Configuration File" on page 30 for more information).

If you answer no, *share_setup* props you to specify the hardware components in one class. To include additional classes in the tree, answer yes when you see this prompt:

```
Are there more client classes using this share tree (y/n)?
```

When you finish specifying the hardware components for all classes, *share_setup* saves the information in the configuration file under the name that you specified in step 7.

10. Verify that your share tree configuration file executes.

See the information in "Verifying the Share Tree Configuration File" on page 36 for this step.

Editing a Share Tree Configuration File

To modify an existing share tree configuration file, you must use a text editor to make your changes. You can also use a text editor instead of *share_setup* to create a new share tree configuration file.

When you install Diskless software on a server, a default version of the configuration file is stored at */var/boot/share.dat*. If you are creating a new configuration file with a text editor, use a working copy of *share.dat* as the source for the new file. (Do not edit the original version of *share.dat*; make a copy of it and make your changes to the copy.) Some values in the default version of *share.dat*, though not all, must be changed to reflect conditions in your diskless implementation.

Refer to these sections for information on setting variables when you edit a working copy of *share.dat*:

- “Server Variables in the Share Tree Configuration File” on page 27
- “Share Tree Variables in the Share Tree Configuration File” on page 29
- “Client Hardware Variables in the Share Tree Configuration File” on page 30

Verifying the Share Tree Configuration File

The share tree configuration file is an executable shell script. After you create a new configuration file or edit an existing one, check its syntax by executing it, as shown in this example:

```
# sh IRIX_65-32.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 the settings in a share tree configuration file; it checks file syntax only.

Installing the Share Tree

To build a share tree, enter the *share_inst* command and the name of the share tree as a command argument. The *share_inst* command automatically calls *inst* to install the software from its installation location. The following example illustrates a *share_inst* command entry. Notice that the *.dat* extension is omitted from the filename of the share tree configuration file:

```
# share_inst -r 32-bitIRIX_6.5
```

See Table 2-2 for a list of subsystems recommended for share tree installations.

Note: The *nfs.diskless_client* subsystem is mandatory.

share_inst Command

The *share_inst* command launches a share tree installation and also deletes a share tree. Syntax for the *share_inst* command is described below (see the man page *share_inst* for a more complete description).

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

The arguments to *share_inst* command have the following meaning.

- | | |
|---------------------|---|
| -r | The flag for the release_name argument. |
| <i>release_name</i> | The name of the configuration file (a working copy of <i>share.dat</i>) without the <i>.dat</i> extension. See “Share Tree Configuration File” on page 26 for details. |
| -d | The flag to delete a share tree. |

This example deletes the share tree *IRIX_62*.

```
# share_inst -r IRIX_62 -d
```

Share Tree Installation Procedure

The procedure below explains how to use *share_inst* to launch a share tree installation. It assumes that you are using Inst, the command line interface to the installation program, but you can use Software Manager, the graphical interface, instead. (See *IRIX Admin: Software Installation and Licensing* for information on using Inst; see the *Personal System Administration Guide* for information on using Software Manager.) The location of the installable software source (local CDs, remotely-mounted CDs, or a remote directory) affects steps 3 and 4 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_65-32
```

This entry directs *share_inst* to read *IRIX_65-32.dat* (the working copy of *share.dat*) to build a *share* tree for 32-bit clients running IRIX 6.5. After it processes your entry, *share_inst* displays a confirmation prompt:

```
About to install shared tree at/diskless/share/IRIX_65-32.....
Enter confirmation (y/Y):
```

Note: The pathname of your share tree should match the pathname that appears in this prompt.

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. open [source...] | Specify additional software locations |
| 3. close [source] | Close distributions |
| 4. list [keywords][names] | Display information about software subsystems |
| 5. go | Perform software installation and removal now |
| 6. install [keywords][names] | Select subsystems to be installed |
| 7. remove [keywords][names] | Select subsystems to be removed |
| 8. keep [keywords][names] | Do not install or remove these subsystems |
| 9. step [keywords][names] | Enter interactive mode for install/remove/keep |

10. conflicts [choice...]	List or resolve installation conflicts
11. help [topic]	Get help in general or on a specific word
12. view ...	Go to the View Commands Window
13. admin	Go to the Administrative Commands Menu
14. 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 remotely mounted 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 reads the products available at that location.

4. Insert the next CD, if needed.

If you have more than one CD to install, enter the *open* command after inserting each additional CD so that its contents can be read:

```
open bigsgi:/CDROM/dist
```

5. Enter **list** to check selected subsystems and space requirements (optional):

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

The *list* option posts the names of software subsystems available from the source; the subsystems that *inst* selects as default installation choices are designated by the letter *i*. The *list* option also calculates the disk space needed to install the selected subsystems.

6. Verify that *nfs.sw.dsk_client* is selected for installation.
7. Change the default selections if necessary.

Note: See Table 2-2 for a list of subsystems suggested for installation.

If you want to change the default selections, use the *step i* option to advance through the list of selected subsystems on the source. Use one of these entries for each subsystem that the *step* option posts:

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.

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

8. Check for conflicts and resolve them, if you have any.

```
Inst> conflicts
```

Use the messages returned by the *conflicts* command to adjust your selections.

9. 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.
```

```
Inst>
```

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

```
Inst> quit
```

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

If there is a problem with the *exports* file (it might be corrupted or in use by another user, for example), you will see a message in your current window and a message is 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 the *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_65-32
# ls
dev/ lib/ sbin/ var/
```

Creating Client Trees

To install software in a client tree, you run the `/var/boot/client_setup` program to create a new client tree configuration file that specifies the requirements of your client and swap trees. To modify a client tree configuration file that already exists, you use a text editor. When the configuration file is finished, you run the client tree installation script, `/var/boot/client_inst`, to install the client and swap trees. Both the client tree configuration file and the share tree configuration file are used in the client tree installation process. After the client and swap trees are installed, clients are ready to be booted and put into service.

This chapter provides the information and procedures that are needed to configure and install client and swap trees. It contains these sections:

- “Client Tree Configuration File” on page 44
- “Creating a New Client Tree Configuration File” on page 52
- “Editing the Client Tree Configuration File” on page 54
- “Verifying the Client Tree Configuration File” on page 54
- “Installing Client and Swap Trees” on page 55
- “Booting a Client Workstation” on page 60

Note: To update an existing client tree with a newer version of the software, use the information in Chapter 5, “Updating Software on Diskless Trees.”

Client Tree Configuration File

The client tree configuration file contains a list of variables that specify the location of the client tree, the location and size of the swap tree, and the NIS domain of the client. This file also specifies the hardware components of the client, because hardware-specific software modules must be installed in each client tree. The software installed in a client tree is obtained from the share tree that supports it.

Note: Your server should already have a default copy of a client tree configuration file in */var/boot/client.dat* as a result of installing Diskless software.

To create a new client tree configuration file, you can run the */var/boot/client_setup* program, or you can create a working copy of the */var/boot/client.dat* file that was stored on the server during the Diskless software installation process. If you plan to modify an existing client tree configuration file, you must use a text editor to make your changes.

Although each client in a diskless implementation must have its own client tree, it is not necessary to create an individual client tree configuration file for each client. Because the information in one configuration file frequently pertains to an entire client class, a single copy of the file can often be used to create multiple client trees. You can use the client tree configuration file for multiple client trees whenever these conditions are true:

- Clients are identical workstation models (including graphics hardware).
- Client trees for all clients is in the same directory.
- Swap trees for all clients is 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 R4000 Elan workstations with client trees in the */diskless/client* directory, swap trees in the */diskless/swap* directory, and 64 MB of swap space.

Client tree configuration files are stored in the */var/boot* directory. By convention, a client tree configuration file is named for the model of clients that it configures. For example, client tree configuration files might have names like this:

```
/var/boot/Indigo.dat  
/var/boot/Indigo_XS.dat  
/var/boot/O2.dat
```

Figure 4-1 illustrates two client configuration files that are used to create client trees for seven clients.

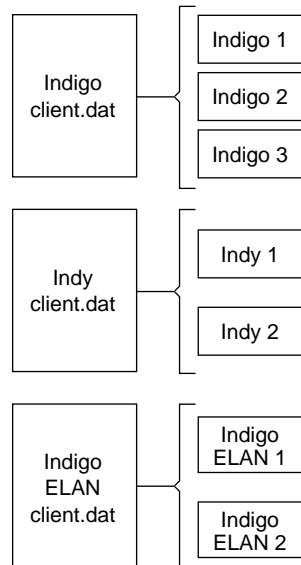


Figure 4-1 Multiple Client Trees From One Configuration File

Client and Swap Tree Variables

The variables in the upper portion of the client configuration file 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 file.

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

Example 4-1 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: 1.30 $
#
#
#       DISKLESS CLIENT TREE SPECIFICATIONS
#
#       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="64m"
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 (in the share tree configuration file); be sure your entry ends in the string *\$HOST*. The value of *\$HOST* is the hostname 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 (in the share tree configuration file); be sure your entry ends in the string *\$HOST*. The value of *\$HOST* is the hostname 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 64 megabytes. If you change this value, your entry must end with the letter *m* (lowercase), 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* on the client so that the client can participate in NIS.

Client Hardware 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/GFXBOARD/MODE values for 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)
#
# NOTES      NOTES      NOTES      NOTES      NOTES
#
# 1. For CPUARCH values which are derivatives of the R4000 architecture,
#    like R4400, just use R4000 as the CPUARCH value.
#
# 2. CPUARCH should be set to the values shown below.
#
# 3. If you modify the supported or unsupported architectures below, please
#    change the file dl_classes.
#
# 4. The MACH variable is no longer applicable to current releases.
#
# 5. The MODE pad should be set on Indigo2 R10000, O2, & OCTANE.
#
#-----+-----+-----+-----+-----+-----+-----+-----+
#| Machine Type      | CPUARCH | CPUBOARD | GFXBOARD | VIDEO |
#|                   |         | /MACH    |          |      |
#-----+-----+-----+-----+-----+-----+
#| 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" |
#| Indigo2 IMPACT    | R4000   | IP22     | "MGRAS -mSUBGR=MGRAS" |
#| Notes: (1) Extreme+XL indicates dual heads: one Extreme, one XL.
#| (2) The galileo board is only a graphics option. It does not change
#|     the architecture of a diskless client.
#

```

```

#| Indy          | R4000 | IP22 | "NEWPORT -mSUBGR=NG1" | VINO
#| Indy R5000    | R5000 | IP22 | "NEWPORT -mSUBGR=NEWTON" | VINO
#|   Note: VIDEO tag is only used by Indy for now.
#|
#| Indigo2 10000
#|   IMPACT      | R10000 | IP28 | "MGRAS -mSUBGR=I2_64"
#|   Extreme     | R10000 | IP28 | "EXPRESS -mSUBGR=I2_64"
#|   XZ          | R10000 | IP28 | "EXPRESS -mSUBGR=I2_64"
#| NOTE: MODE="64bit"
#|
#| OCTANE
#|   SI/SSI/MXI | R10000 | IP30 | "MGRAS -mSUBGR=RACER"
#| NOTE: MODE="64bit"
#|
#| O2            | R5000  | IP32 | "CRIME -mSUBGR=CRM" | VICE
#|              | R10000 | IP32 | "CRIME -mSUBGR=CRM" | VICE
#| NOTE: MODE="32bit"
#|
+-----+-----+-----+-----+-----+
#
# The following machines are NOT supported for diskless operation in the
# current release. The last column states the version of IRIX which last
# supported a specific architecture. NEVER means a specific architecture
# was never supported in a diskless configuration.
#
+-----+-----+-----+-----+-----+
#| Machine Type | CPUARCH | CPUBOARD | GFXBOARD | VERSION|
+-----+-----+-----+-----+-----+
#| 4D/50G       | R3000   | IP4      | "CLOVER1" | NEVER
#| 4D/50GT      | R3000   | IP4      | "CLOVER2 -mSUBGR=IP4GT" | NEVER
#| 4D/70G       | R3000   | IP4      | "CLOVER1" | NEVER
#| 4D/70GT      | R3000   | IP4      | "CLOVER2 -mSUBGR=IP4GT" | NEVER
#|
#| 4D/120GTX    | R3000   | IP5      | "CLOVER2 -mSUBGR=IP5GT" | NEVER
#|
#| 4D/xy0GTX    | R3000   | IP{5,7,9} | "CLOVER2 -mSUBGR=IP7GT" | NEVER
#| 4D/xy0RE     | R3000   | IP{5,7,9} | "VENICE -mSUBGR=IP7" | NEVER
#| 4D/xy0SKY    | R3000   | IP{5,7,9} | "STAPUFT -mSUBGR=SKYWR" | NEVER
#| 4D/xy0VGX    | R3000   | IP{5,7,9} | "STAPUFT -mSUBGR=IP7GT" | NEVER
#|   where x=2, 3 or 4, and y=1, 2, 4, or 8.
#|
#| 4D/20{G,TG}  | R3000   | IP6      | "ECLIPSE -mSUBGR=ECLIPSE" | 5.3
#| 4D/25{G,TG}  | R3000   | IP6      | "ECLIPSE -mSUBGR=ECLIPSE" | 5.3
#| Key: 4D/20{G,TG} is an abbreviation for 4D/20, 4D/20G, and 4D/20TG.
#|

```

#	4D/210GTX				
#	4D/30{G,TG}	R3000	IP12	"ECLIPSE -mSUBGR=ECLIPSE"	5.3
#	4D/30EG (Elan)	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#	4D/30XS{24}	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#	4D/35XS{24}	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#					
#	4D/35{G,TG}	R3000	IP12	"ECLIPSE -mSUBGR=ECLIPSE"	5.3
#	4D/35EG (Elan)	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#					
#	Indigo	R3000	IP12	"LIGHT -mSUBGR=LIGHT"	5.3
#	Indigo Dual Head	R3000	IP12	"LIGHT -mSUBGR=LIGHT"	5.3
#	Indigo XS{24}	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#	4D/35EG (Elan)	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#					
#	Indigo	R3000	IP12	"LIGHT -mSUBGR=LIGHT"	5.3
#	Indigo Dual Head	R3000	IP12	"LIGHT -mSUBGR=LIGHT"	5.3
#	Indigo XS{24}	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#	Indigo Elan	R3000	IP12	"EXPRESS -mSUBGR=EXPRESS"	5.3
#					
#	Crimson BLG	R4000	IP17	"LIGHT -mSUBGR=IP17"	NEVER
#	Crimson XS{24}	R4000	IP17	"EXPRESS -mSUBGR=IP17"	NEVER
#	Crimson Elan	R4000	IP17	"EXPRESS -mSUBGR=IP17"	NEVER
#	Crimson GTX	R4000	IP17	"CLOVER2 -mSUBGR=IP17"	NEVER
#	Crimson VGX	R4000	IP17	"STAPUFT -mSUBGR=IP17"	NEVER
#	Crimson VGXT	R4000	IP17	"STAPUFT -mSUBGR=IP17SKY"	NEVER
#	Crimson RE	R4000	IP17	"VENICE -mSUBGR=IP17"	NEVER
#					
#	Onyx				
#	Extreme	R4000	IP19	"EXPRESS -mSUBGR=IP19"	NEVER
#	Extreme	R10000	IP25	"EXPRESS -mSUBGR=IP25"	NEVER
#	RealityEngine2	R4000	IP19	"VENICE -mSUBGR=IP19"	NEVER
#	RealityEngine2	R10000	IP25	"VENICE -mSUBGR=IP25"	NEVER
#	InfiniteReality	R4000	IP19	"KONA -mSUBGR=IP19"	NEVER
#	InfiniteReality	R10000	IP25	"KONA -mSUBGR=IP25"	NEVER
#					
#	PowerOnyx				
#	Extreme	R8000	IP21	"EXPRESS -mSUBGR=IP21"	NEVER
#	RealityEngine2	R8000	IP21	"VENICE -mSUBGR=IP21"	NEVER
#	InfiniteReality	R8000	IP21	"KONA -mSUBGR=IP21"	NEVER
#					
#	Power Indigo2				
#	Extreme	R8000	IP26	"EXPRESS -mSUBGR=I2_64"	6.0.1
#	XZ	R8000	IP26	"EXPRESS -mSUBGR=I2_64"	6.0.1
#					
#					

```
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.
- 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** Specifies the video chip for Indy (VINO) and O2 (VICE) workstations. The default setting is null, which indicates systems with no on-board video hardware.
Set *VIDEO* to *VINO* for Indy workstations; set it to *VICE* for O2 workstations. For other models, use the default setting (null).
- MODE** Specifies the 32-bit or 64-bit version of the operating system.

Creating a New Client Tree Configuration File

Use the *client_setup* setup program to create a client tree configuration file for an individual client or class of clients in your diskless implementation (see “Client Tree Configuration File” on page 44).

Note: If you prefer to use an editor to create this file, or if you need to modify an existing client tree configuration file, use the information in “Editing the Client Tree Configuration File” on page 54 instead of this procedure.

This procedure explains how to create a new client tree configuration file using *client_setup*.

1. Start *client_setup*.

```
# cd /var/boot
# ./client_setup
```

After your entry, this prompt is displayed:

```
This program will help you create a client tree configuration
file for the client you would like to install.
```

```
Choose the CPU board:
```

- 1. IP20
- 2. IP22
- 3. IP28
- 4. IP30
- 5. IP32
- 6. IP35

```
Choose one (1 2 3 4 5 6 ):
```

2. Enter the number of your selection.

After your entry, your selection is confirmed and another prompt is displayed:

```
>> IP20 selected.
```

```
Choose the graphics board:
```

- 1. LIGHT
- 2. EXPRESS

```
Choose one (1 2):
```

3. Enter the number of your selection.

After your entry, your selection is confirmed and another prompt is displayed:

```
>>LIGHT selected.
```

You have defined a client class with these components:

```
CPUBOARD=IP20
CPUARCH=R4000
GFXBOARD=LIGHT -mSUBGR=LG1MC
VIDEO=
MODE=32bit
```

Specify a size in MB for swap.

Swap size (Default is 64m - <Enter> for default):

4. Specify the size of your swap file.

If you wish to specify a swap file that is different from the default, enter the size (the unit of measure is megabytes). Or, press **Enter** to set the swap size to 64 megabytes.

After your entry, this prompt is displayed:

Enter your NIS domain name (hit <Enter> for no domain name):

5. Specify a domain or simply press **Enter**.

If the clients supported by this file are using an NIS domain name that is different from the server's, enter the domain name for the clients. Otherwise, press **Enter** to set the client domain name to the domain name of the server.

After your entry, this prompt is displayed:

Specify a name for your client tree configuration file
(no extension please):

6. Specify a name for this client tree configuration file.

See "Client Tree Configuration File" on page 44 for naming conventions.

7. Verify that your client tree configuration file executes.

See the information in "Verifying the Client Tree Configuration File" on page 54 for this step.

Editing the Client Tree Configuration File

To modify an existing client tree configuration file, use a text editor to make your changes. You can also use a text editor instead of *client_setup* to create a new client tree configuration file.

When you install Diskless software on a server, a default version of the configuration file is stored at */var/boot/client.dat*. If you are creating a new configuration file with a text editor, use a working copy of *client.dat* as the source for the new file. (Do not edit the original version of *client.dat*; make a copy of it and make your changes to the copy.) You will need to change several values in the default version of *client.dat*.

Refer to these sections for information on setting variables when you edit a working copy of *client.dat*:

- “Client and Swap Tree Variables” on page 46
- “Client Hardware Variables” on page 48

Verifying the Client Tree Configuration File

The *client.dat* file and its working copies are executable shell scripts. After you create a new client tree configuration file or edit an existing one, 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 Client and Swap Trees

To install a client tree, run *client_inst*, which reads the working copy of the client and share tree configuration files for configuration information; then *client_inst* calls *inst* to install the software. Because the share tree serves as the software source, you must have installed the share tree before installing any of the client trees that will use it. After you create one client tree, you can use the *client_clone* program to create additional copies of the tree (see “Cloning Client Trees” on page 59). Individual client trees are stored in */diskless/client_root/client_name*.

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

As part of the client tree installation process, *client_inst* installs the swap tree for each client in */diskless/swap/client_name*. It also creates the */etc/fstab* and */etc/hosts* files that clients need and includes them in the client trees. Finally, *client_inst* adds entries that clients need 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 “Assigning Internet Addresses” in Chapter 2 for details).

The *client_inst* Command

The *client_inst* command initiates client and swap tree installations. This command requires arguments that identify the hostname of the client being installed and the share tree and client tree configuration files to be used in the installation.

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

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

Arguments to *client_inst* have these meanings:

- h** *hostname* Specifies the hostname of an individual client whose tree you are installing.
- r** *release* Specifies the share tree configuration file (without the *.dat* extension) that is used during client tree installations.
- c** *class* Specifies the working copy of *client.dat* for this class (without the *.dat* extension).
- d** Deletes a client tree.

The following sample illustrate *client_inst* command entries.

```
client_inst -h starlite -r IRIX_65-32 -c O2
```

The entry in this sample installs a client tree and swap tree for a client named *starlite* using a share tree configuration file called *IRIX_65-32* (the *.dat* extension is omitted) and a client tree configuration file named *O2* (the *.dat* extension is omitted).

Client and Swap Tree Installation Procedure

The procedure in this section explains how to use *client_inst* to launch a client tree installation that creates a client and swap tree. Remember, only software that is installed on the share tree can be installed in a client tree.

When running *client_inst*, use the same media in the same order that you used with *share_inst*, that is, the operating system CD first, then the NFS CD, and any other source media.

The procedure assumes that you are using *Inst*, the command-line interface to the installation program, but you can use *Software Manager*, the graphical interface, instead. (See *IRIX Admin: Software Installation and Licensing* for information on using *Inst*; see the *Personal System Administration Guide* for information on using *Software Manager*.) After a client tree is created, you can add new clients by cloning the tree (see “Cloning Client Trees” on page 59).

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. Shut down the client system.
2. From the `/var/boot` directory, start `client_inst`, use the same media in the same order that you used with `share_inst`, that is, the operating system CD first, then the NFS CD, and any other source media.

```
# ./client_inst -h starlite -r IRIX_65-32 -c O2
```

This example directs `client_inst` to create a client tree for `starlite` using configuration information in `IRIX_65-32.dat` for its share tree and `O2.dat` for its client tree. A swap tree is automatically created for this client.

After processing this entry, `client_inst` displays this prompt:

```
Client tree = /diskless/client/starlite,
              shared tree = /diskless/share/IRIX_65-32
Enter confirmation (y/Y):
Create 64m swap file .....
```

Note: The pathname of the client and share trees in this prompt should match the pathnames that appear in this prompt.

3. 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. open [source...]	Specify additional software locations
3. close [source]	Close distributions
4. list [keywords][names]	Display information about software subsystems
5. go	Perform software installation and removal now
6. install [keywords][names]	Select subsystems to be installed
7. remove [keywords][names]	Select subsystems to be removed
8. keep [keywords][names]	Do not install or remove these subsystems
9. step [keywords][names]	Enter interactive mode for install/remove/keep
10. conflicts [choice...]	List or resolve installation conflicts
11. help [topic]	Get help in general or on a specific word
12. view ...	Go to the View Commands Window
13. admin	Go to the Administrative Commands Menu
14. quit	Terminate software installation

Inst>

4. Enter **list** to check to check the software selected for installation (optional).

The *list* option displays the contents of the share tree; the subsystems selected for installation (automatically chosen by *inst*) are marked by the letter *i*.

The *list* option also calculates the disk space needed to install the default selections and displays the amount of disk space that is available for installation. When the calculations are complete, check the disk space summary to determine the amount of disk space the client tree will require.

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

This subsystem is required for diskless operation.

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

6. Change the default selections if necessary.

If you want to change the default selections, use the *step i* option to advance through the list of selected subsystems on the source. Use one of these entries for each subsystem that the *step* option posts:

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.

7. 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.
```

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

After your entry, *inst* processes a suite of *exit* commands. Allow the *exit* commands several minutes to process. During exit processing, watch for a “building kernel” message.

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. You can repeat this procedure for other clients, or use the *clone_client* command to reproduce additional client trees automatically (see “Cloning Client Trees” on page 59).

Cloning Client Trees

Client trees can be reproduced using the *clone_client* program. Like the conventional *client_inst* process, *clone_client* automatically creates a swap tree when it creates a client tree. When you use *clone_client*, you can create a duplicate client tree from another client tree or from the share tree. The software installed in cloned trees is obtained from the same source as the software used to install the model client or share tree originally. For example, if the software in the model client tree */diskless/client/starlite* was installed from the remote distribution directory *sandy:/dist/IRIX65-32*, then the software in any tree cloned from the starlite tree is obtained from that directory on sandy.

The *clone_client* command has this syntax:

```
clone_inst -f filename -r release -c class -clone src_root
```

Specify this information in command arguments:

- f filename** A file containing the names of hosts whose trees are being created by the cloning process.
- r release** The working copy of the share tree configuration file that defines this client’s share tree (without the *.dat* extension).
- c class** Is the name of the working copy of client tree configuration file that was used for the model client tree (without the *.dat* extension).
- clone src_root** The complete pathname to the source of the software to be included in client tree. The source can be a share tree or another client tree. You must specify the complete pathname of the source. For example, if starlite’s client tree is used as the model, this argument will read *-clone /diskless/client/starlite*.

Booting a 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: Be sure that `bootparams` is enabled in the server's `/etc/inetd.conf` file before you begin this procedure (see step 9 of "Checking a Server" on page 8).

Go to the client workstation to do this procedure:

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

If the client is running, type `shutdown -y -g1` on the console to bring down the system. If you do not shut down the client, a message warns you that the client might crash.

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 R4000 workstations 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
>>setenv SystemPartition bootp()sandy:/var/boot/starlite
>>setenv OSLoadPartition bootp()sandy:/var/boot/starlite
>>setenv OSLoader /unix
```

The variables in these examples have the following meaning:

<i>diskless</i>	The variable that enables diskless service. Set <i>diskless</i> to 1.
<i>netaddr</i>	The client's Internet address. See "Assigning Internet Addresses" in Chapter 2 if you need help with this setting.
<i>bootfile</i>	The complete pathname for the client's boot file. In the sample command above, <i>/var/boot/starlite/unix</i> on server <i>sandy</i> will be used to boot client <i>starlite</i> .
<i>SystemPartition</i>	The directory where the operating system loader is located. In the previous sample command, the directory <i>/var/boot/starlite</i> on server <i>sandy</i> contains the operating system loader for <i>starlite</i> .
<i>OSLoadPartition</i>	The directory where the core operating system is located. In the previous sample command, the directory <i>/var/boot/starlite</i> on server <i>sandy</i> contains the boot file for <i>starlite</i> .
<i>OSLoader</i>	is the name of the IRIX boot file. The previous sample command specifies the boot file to be <i>/unix</i> .

Note: IP35-based systems such as Tezro, Fuel, Onyx350, and Origin350/3000 support the following additional prom variables: *netmask*, *dlgate* (download gateway), *dlserver* (download server).

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** from the Command Monitor.

8. Configure the client as you would normally configure any host.

For many networks, host configuration commonly includes such tasks as setting a netmask, mounting (or automounting) filesystems, and setting up user accounts. Perform such tasks on the diskless clients that you are running.

9. Reboot each client to put them into operation.

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).

Updating Software on Diskless Trees

This chapter explains how to update the software for 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” on page 63
- “Updating a Share Tree” on page 64
- “Updating Existing Client Trees” on page 67
- “When New Clients Need Updated Trees” on page 68
- “Software Options as Updates” on page 68

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.

Caution: If you do not shut down clients to perform an update, they might crash during the update or 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.

Because software was previously installed in the tree that you are updating, *inst* has a record of the installation; it uses this record to identify currently installed software for which the new software contains a revision. In *inst* listings, these subsystems are marked with the letters *iU*, indicating that they are updates and are selected for installation.

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 on the server.

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

3. Start *share_inst*.

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

```
# ./share_inst -r IRIX_65-32
```

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

```
WARNING: release IRIX_65-32 is still serving clients.  
continue to update IRIX_65-32 (Y/N)?
```

If you answer 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_65-32.....  
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

1. from [source...]	Specify location of software to be installed
2. open [source...]	Specify additional software locations
3. close [source]	Close distributions
4. list [keywords][names]	Display information about software subsystems
5. go	Perform software installation and removal now
6. install [keywords][names]	Select subsystems to be installed
7. remove [keywords][names]	Select subsystems to be removed
8. keep [keywords][names]	Do not install or remove these subsystems
9. step [keywords][names]	Enter interactive mode for install/remove/keep
10. conflicts [choice...]	List or resolve installation conflicts
11. help [topic]	Get help in general or on a specific word
12. view ...	Go to the View Commands Window
13. admin	Go to the Administrative Commands Menu
14. 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 default location, press the Enter key and go to Step 6.

Note: If you used CD-ROM as the source for the previous installation, insert the CD before pressing the Enter key.

- If you plan to install the updates from a location other than the default, specify the location with the **from** option, insert a CD if one is required, and press Enter, then go to Step 6.

6. Enter **list** to determine the default installation selections.

The *inst* utility automatically identifies software updates and selects them for installation. By default, *inst* selects all new subsystems (designated by the letters *i N*) and all subsystems that are updates (designated by the letter *i U*). Use the *list* option to check these selections and their space requirements before installing them.

7. Change the default selections if necessary.

If you want to change the default selections, use the *step i* option to advance through the list of selected subsystems on the source. Type one of these entries for each subsystem that the *step* option posts:

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.

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

When your selections are satisfactory, enter the *go* option 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 eoe.sw.xfs subsystems
Installing new versions of selected eoe.sw.xfs subsystems

Removing old versions of selected desktop_eoe.sw.envm subsystems
Installing new versions of selected desktop_eoe.sw.envm subsystems
```

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

```
Installation and/or removal succeeded.
```

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 processing of the *exit* commands (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 Existing Client Trees

The *clone_client* program for client trees automatically installs updated software from the share tree into an existing client tree. This program installs all subsystems that are new to the client tree and all subsystems for which the share tree contains a newer version. It also removes any subsystem on the client for which there is no corresponding subsystem in the share tree—these subsystems are considered obsolete.

Note: If you are adding a new client to the diskless implementation and you want its client tree to contain updated software, you must complete Step 2 of this procedure twice. (See “When New Clients Need Updated Trees” on page 68 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, *clone_client* warns you that it might crash during the update.

2. Use *clone_client* to launch an automatic update of client trees.

When you use *clone_client* to update a client tree, *inst* takes these actions:

- Installs all new subsystems. This software is on the share tree but is not yet installed on the client tree; these subsystems are marked N in *inst* listings.
- Installs all subsystems in the client tree for which there is a newer version in the share tree; these subsystems are marked U in *inst* listings.
- Removes any subsystems from the client tree that do not exist in the share tree; these subsystems are considered obsolete.

If you prefer to select the software to be updated, complete the instructions in “Client and Swap Tree Installation Procedure” on page 56 instead of using *clone_client* program.

3. Reboot the client workstation.

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

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

When New Clients Need Updated 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. Because no record exists on a new client tree, *inst* installs a baseline version of the 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.

Software Options as Updates

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

Adding Options to a Share Tree

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

1. Enter the *share_inst* command to install software on a share tree.

Your entry will look like the example that follows. In this example, 32bitIRIX_65 is the share tree to which the option is being added:

```
# /var/boot/share_inst -r 32bitIRIX_65
```

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

1. from [source...]	Specify location of software to be installed
2. open [source...]	Specify additional software locations
3. close [source]	Close distributions
4. list [keywords][names]	Display information about software subsystems
5. go	Perform software installation and removal now
6. install [keywords][names]	Select subsystems to be installed
7. remove [keywords][names]	Select subsystems to be removed
8. keep [keywords][names]	Do not install or remove these subsystems
9. step [keywords][names]	Enter interactive mode for install/remove/keep
10. conflicts [choice...]	List or resolve installation conflicts
11. help [topic]	Get help in general or on a specific word
12. view ...	Go to the View Commands Window
13. admin	Go to the Administrative Commands Menu
14. quit	Terminate software installation

Inst>

2. Enter **from** to specify a different software location, if necessary.

The *inst* utility assumes that you are installing the software option from the same location as the previous installation (the "default location" message above the Main Menu specifies CD, because 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* online help if you need instructions.

3. Enter **keep I** 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 I** entry tells *inst* to ignore all subsystem selections except those that are currently installed.

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:/d2/32bitIRIX65
```

4. Enter `install option_name` to select the software option.
For example, if you want to install the MineSet product option, your entry would look like this:

```
inst> install MineSet
```
5. Enter `list` to verify the software that you selected:

```
Inst> list
```
6. Enter `go` to launch the installation:

```
Inst> go
```
7. 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 Client Trees

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_65-32 -c O2 -h starlite
```

In this example, *starlite*, the client using the IRIX_65-32 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 I` to clear unwanted subsystem selections.
Because many subsystems in the share tree are selected for installation, you need to clear these selections before going on with the installation. The `keep I` entry tells `inst` to ignore all subsystem selections except those that are currently installed.
3. Enter `install` to select the software that you want to add.

```
Inst> install MineSet
```


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.

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” on page 74
- “Removing a Share Tree” on page 75
- “Changing the Size of the Swap Space” on page 76
- “Setting Up a Local Disk” on page 76
- “Moving a Client to a Different Server” on page 79
- “Converting a Client to a Standalone Workstation” on page 83

Removing a Client Tree

To remove a client tree, 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 argument, as shown in this sample entry, which removes host `starlite`:

```
# /var/boot/client_inst -h starlite -r IRIX_62 -c Indigo_XS -d
```

After your entry, you see a response similar to this:

```
Remove client tree at /diskless/client/starlite
                        (shared tree = /diskless/share/IRIX_62
Enter confirmation (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)?
```

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 hostname 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, because removing a share tree disables its clients. You should also make a backup copy of the share tree before starting this procedure, just in case you need to restore it later.

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_62 -d
About to remove shared tree at /diskless/share/IRIX_62...
Enter confirmation (y/Y):
```

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

After your entry, wait 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_62 is still serving clients.
continue to delete IRIX_62(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 "Client and Swap Tree Variables" in Chapter 4). The default size is 64 megabytes. 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 100 megabytes:

```
# mkfile 100m _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 workstation 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 for 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 *IRIX Admin: Disks and Filesystems* and the *fx* man page for instructions on disk repartitioning.

Getting Prepared

To prepare for setting up resources on the local disk, first 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, enter the *hinv* command and check the output 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 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 filename) 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	<i>root</i> partition
1	<i>swap</i> partition
6	<i>usr</i> 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 */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 as the superuser 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* for more information). This entry mounts the */tmp* filesystem on the root partition:

```
/dev/dsk/dks0d1s0 /var/tmp xfs rw,raw=/dev/rdisk/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 swap partition on a local (system) disk using an */etc/fstab* file entry in a client tree. This swap partition is 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 in this entry (*pri=3*) 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 hostname, 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_65-32 -h starlite
```

If you need instructions to complete this step, see “Client and Swap Tree Installation Procedure” on page 56.

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 backup media. 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 removable media, 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 must also be changed. See Steps 4 and 5 of “Booting a Client Workstation” on page 60 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 (if any) 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. If the operating system is not installed on the disk, refer to *IRIX Admin: Software Installation and Licensing*.

1. Follow all instructions in "Preparing the Client" on page 80.
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:

```
>> resetenv
```

9. Initialize the workstation environment:

```
>> setenv netaddr xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* is the network address of the workstation.

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 in "Enabling Network Software" on page 85 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 hostname 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 hostname in */etc/sys_id*.

Edit the */etc/sys_id* file and enter the hostname 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 the DNS or NIS name service, the service needs a record of the new hostname and address. If you do not administer the name service, notify its administrator to add the new host to its database.

If your site does not use a name service, 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**.

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 80 for instructions.

2. Remove the client tree from the server.

See “Removing a Client Tree” on page 74 for instructions.

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

See “Removing a Share Tree” on page 75 for instructions.

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 that one is necessary.

This chapter contains these sections:

- “General Approach to Troubleshooting” on page 87
- “Troubleshooting Checklist” on page 88
- “Installation Error Messages” on page 89
- “Removing a Diskless Class Manually” on page 90
- “Debugging the Boot Process” on page 91
- “Before You Call for Help” on page 94

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:

- Diskless software is improperly configured.
- The boot process is not completing successfully.

Troubleshooting Checklist

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

1. Did you install an official IRIX release that both the client and server support?

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 because *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, because *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*, *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
<code>/var/boot/classname.dat not found</code>	The <i>client.dat</i> file for the share tree that <i>client_inst</i> is trying to build is missing. Check <code>/var/boot</code> to make sure that the appropriate <i>client.dat</i> is listed and check your spelling of the file name.
<code>Invalid hostname_clientname</code>	<i>client_inst</i> 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 <code>/etc/hosts</code> 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.
<code><clientname> is not a diskless client for CLASS classname</code>	You are trying to remove a client from the wrong diskless server or <i>client_inst</i> cannot determine the client-server relationship. If NIS=no in the configuration file for this client, check the server's local <code>/etc/bootparams</code> file for correct entries. If NIS=yes in the configuration file for this client, check the NIS <i>bootparams</i> 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:/dl/client/dayglo \
    sbin=babylon:/dl/share/5_1/sbin swap=babylon:/dl/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. The four phases of the boot process are 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 a broadcast going to all servers on the same network, or a specific request to a known diskless server. If a router is installed between the server and its client, the router must be configured to forward *bootp* requests (see the *-f* argument to *bootp* in the *bootp(1M)* reference page). The transport mechanism is the User Datagram Protocol (UDP/IP).

Once the *bootp* request reaches the diskless server, the super server *inetd* 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 filename, which is specified in the client's PROM variable, *bootfile*.

Phase 2: The Boot File Download

When the *bootp* server has resolved the client identity and boot file location, the client sends a *tftpd* (the Trivial File Transport Protocol) request for its boot file, */var/boot/client_name/unix*, which invokes TFTP on the server. The server sends the kernel and the client loads it into memory.

Phase 3: The Server-Client Setup

Once the client has loaded the kernel into memory, it initiates a broadcast RPC 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*, 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, the client issues an NFS request to the server for its *root*, *sbin*, and *swap* filesystems. The server then starts the *rpc.mountd* server to answer the request (*rpc.mountd* is responsible for verifying access to a filesystem by clients and users). 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.

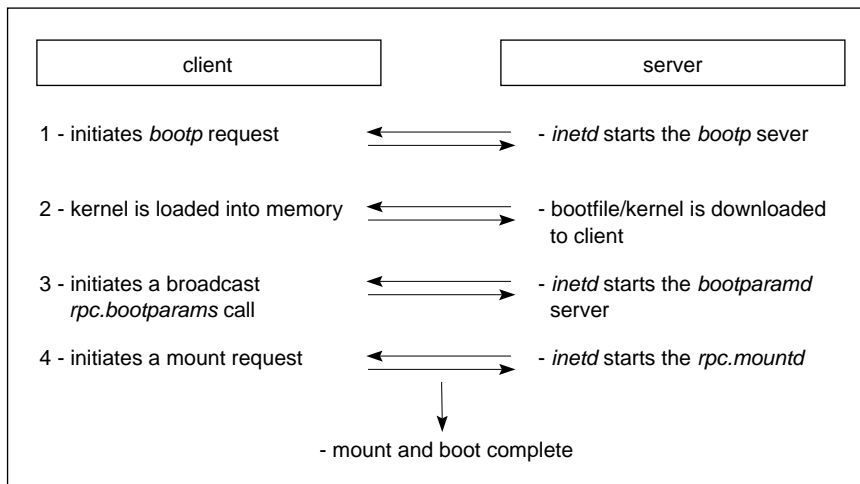


Figure 7-1 Diskless Boot Process

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	What to Check
<i>[ec0, enp0, et0, ipg0, fxp0]: transmit: no carrier ef, me no server for server: /var/boot/client/unix Unable to continue: press <enter> to return to the menu:</i>	Check the network cable to be sure that it is securely attached.
<i>no server for server: /var/boot/client/unix Unable to continue: press <enter> to return to the menu:</i>	On the client, check the NVRAM <i>netaddr</i> variable to ensure that it has the correct Internet address. On the server, check the <i>/etc/inetd.conf</i> file to ensure that the <i>bootp</i> server is supported.
<i>File /var/boot/client/unix not found on server servername Unable to continue: press <enter> to return to the menu:</i>	On the client, the NVRAM <i>bootfile</i> variable might have the wrong path or bootfile name. On the server, check to ensure that the bootfile specified in the <i>bootfile</i> variable on client actually exists.
<i>Starting up the system ... Copyright message... PANIC: KERNEL FAULT.....</i>	On the client, the <i>diskless</i> 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 <i>client.dat</i> and <i>server.dat</i> to be sure the correct kernel modules have been specified.
<i>Starting up the system... Unable to continue: press <enter> to return to the menu:</i>	On the server, check the <i>/etc/inetd.conf</i> file to ensure that the <i>tftpd</i> server is supported and configured correctly. The default <i>tftpd</i> entry runs in secure mode and supports the diskless environment. The default entry is <code>tftpd -s /var/local/boot</code>
<i>kernel mount failed, check server, bootparams or press reset button!!! Get_bootp failed</i>	On the server, check the <i>/etc/inetd.conf</i> file to ensure that <i>bootparamd</i> is supported (the <i>bootparamd</i> entry is present and is not preceded by the comment character, #).

Table 7-2 (continued) Boot Process Troubleshooting Hints

Client Error Messages	What to Check
<p><i>GET_BOOTP: WHOAMI fail, addr=0x</i> <i>Get_bootp failed</i> <i>Kernel mount failed, check server, bootparams</i> <i>or press reset button!!!</i></p>	<p>On the server or NIS master, check that the <i>/etc/bootparams</i> file contains valid entries for the client. If the client was created with NIS=yes in <i>client.dat</i>, the problem is with the NIS bootparams database. If the client was created with NIS=no in <i>client.dat</i>, the problem is with the server's local <i>/etc/bootparams</i> file. Verify that <i>/var/yp/ypdomain</i> contains an entry for the server.</p>
<p>svl: missing Portmapper not responding; still trying</p>	<p>On the server, check the <i>HOSTNAME</i> setting in <i>share.dat</i> to be sure that it specifies the network interface to the diskless LAN. If the diskless LAN is on a secondary network interface, <i>HOSTNAME</i> should not be set to <i>hostname</i> unless <i>/etc/sys_id</i> contains the name of the diskless LAN interface.</p>
<p><i>Kernel mount failed, check server, bootparams</i> <i>or press reset button!!!</i> <i>Diskless root mount failed:</i></p>	<p>On the server, check the <i>/etc/exports(1M)</i> file. The diskless tree filesystems (<i>root</i>, <i>swap</i>, and <i>share</i>) may not be accessible to clients.</p>
<p><i>nfs_rmount: SHARE MOUNT FAILED.</i></p>	<p>On the server, check the <i>/etc/exports</i> file. The share tree for this client may not be accessible.</p>

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 hostnames.

Index

A

architecture variables. *See* hardware variables.
assigning internet addresses, 13

B

backing up client trees, 80
boot file name resolution, 91
bootfile NVRAM variable, 61, 82, 91
booting
 clients, 60-62
 detailed description of, 91
 troubleshooting, 91-94
bootparamd server, 92
bootparams map, 28
bootp protocol, 91

C

chkconfig command, 85
clearing NVRAM variables, 84
client_inst command
 deleting client trees with, 74
 errors from, 89
 invoking, 56, 57
 overview of, 21
 sample entries, 56
 syntax of, 55, 59

client classes, 6, 44
client.dat file. *See* client tree configuration file.
client name
 setting, 28
clients
 boot file for, 91
 booting, 60-62
 checking, 11
 converting to standalone stations, 83-86
 determining internet address on, 84
 domain name on, 47
 /etc/fstab file for, 21
 hardware components in, 12
 internet address on, 61
 moving, 79-82
 network location of, 11
 shutting down, 63, 80
 verifying operation of, 82
client tree configuration file
 checking syntax of, 54
 client tree variables in, 46-47
 creating, 52
 editing, 51, 54
 hardware variables in, 48-??
 number required, 44
 overview of, 21
 purpose of, 44
 swap tree variables in, 46-47
client trees
 adding options to, 70
 backing up, 80
 cloning, 59

- configuring, 21
- contents of, 6, 17
- creating, 21
- installation procedure, 56-59
- installing, 55-59
- planning, 17
- removing, 82, 86
- removing manually, 90
- root directory of, 14
- space requirements for, 17
- updating, 67
- when to build, 22

cloning

- definition of, 22
- procedure for, 59

CLROOT variable, 47

compatibility (of IRIX versions), 9, 63

CPUARCH variable, 32, 51

CPUBAORD variable, 51

CPUBOARD variable, 32, 88

D

/dev/dsk directory, 77

device special files, 77

disk drives

- configuring, 83

diskless NVRAM variable, 61, 84

diskless tree

- and filesystems, 14
- definition of, 4
- elements of, 4
- planning, 14-20
- removing manually, 90
- root directory of, 14
- space requirements for, 8

DISKLESS variable, 29, 47

disk partitions, 77

disk space

- for client trees, 17
- for diskless tree, 8
- for share trees, 15
- for swap trees, 19

distribution software, 8

domain name

- format of, 10
- of client, 47
- setting, 10

domainname command, 10

E

/etc/bootparams file, 21, 28, 90, 92, 94

/etc/exports file, 10, 14, 21, 90, 92, 94

/etc/fstab file, 21, 55, 81

/etc/hosts file, 10, 28

/etc/inetd.conf file, 9

/etc/sys_id file, 85

/etc/xtab file, 92

export conflicts, 10, 14

exported filesystems, 3

F

filesystems

- and the diskless tree, 14
- in a share tree, 15
- in client trees, 17
- on local disks, 76-78
- using multiple, 15

G

GFXBOARD variable, 32, 51, 88

guest account, 9

H

hardware-specific modules, 30
 hardware variables. *See also individual variable names*
 in client tree configuration file, 48-51
 in share tree configuration file, 30
hinv command, 77, 84
 host database, 10, 28, 74
HOSTNAME variable, 28

I

inetd server, 9, 91, 92
 installation procedure
 summary of, 22
 installation software requirements, 8
inst utility
 and adding software options, 69-70
 and client tree installations, 57-58
 and diskless tree installations, 21
 and new client tree updates, 68
 and share tree installations, 38-41
 and updating share trees, 64-66
 internet address
 assigning, 13
 determining on clients, 84
 of clients, 13, 61, 74
 IRIX
 configuring kernel for, 82
 disk partitions, 77
 distribution media, 8
 on servers, 63
 on share trees, 15, 88
 version compatibility, 9, 63, 88

L

lboot command, 88
 local disks
 adding filesystems to, 76-78
 adding swap space to, 76, 79
 installing, 83
 uses for, 3

M

MACH variable, 88
 mounted filesystems (NFS), 3
 moving clients, 79-82

N

netaddr NVRAM variable, 61, 82, 84
 Network File System. *See* NFS.
 network information service. *See* NIS.
 network interfaces (multiple), 28
 network problems
 troubleshooting, 87
 network software
 enabling, 85
 turning on, 84
 NFS
 administering, 14, 16
 export conflicts, 10, 14
 function in diskless networks, 3
 NIS
 hosts database, 74, 85
 setting domain names for, 9
NISDOMAIN variable, 47
 NVRAM variables
 clearing, 84
 setting, 60, 82, 93

O

- option disks, 76, 78
- options (software)
 - adding to client trees, 70
 - adding to share tree, 68-70

P

- planning client trees, 17
- planning share trees, 15-17
- planning swap space, 19
- printenv* command, 60, 84
- private files
 - backing up, 80
 - copying, 86
 - in client tree, 6
 - in client trees, 19
 - moving, 81, 86

R

- removing client trees, 82, 86
- removing diskless software, 90
- removing share trees, 75
- rpc.mountd* server, 92

S

- server
 - checking suitability of, 8-10
 - cleaning, 82
 - definition of, 1
 - disk space on, 8
 - IRIX version on, 9, 63, 88
 - network location of, 11
 - reliability requirements, 8

SERVER_NIS variable, 28

- server name
 - setting, 28
- setting NIS domain names, 10
- share_inst* command
 - invoking, 38
 - overview of, 20
 - removing share trees with, 75
 - syntax of, 37

share.dat file. *See* share tree configuration file.

SHAREHOST variable, 28

- share tree
 - adding options to, 68-70
 - checking contents of, 41
 - configuring, 20
 - contents of, 5, 15
 - creating, 20
 - hardware-specific modules in, 30
 - installation procedure, 38
 - installing, 37
 - planning, 15-17
 - removing, 75
 - removing manually, 90
 - root directory of, 14
 - selecting software for, 37
 - space requirements for, 15
 - updating, 64-66
 - when to build, 22
- share tree configuration file
 - checking syntax of, 36
 - client hardware variables in, 30
 - editing, 33
 - naming, 26
 - overview of, 20
 - server environment variables in, 27
 - setting variables in, 27-33
 - share tree variables in, 29
- share tree variables (setting), 29
- SHARE* variable, 29

shutting down clients, 63, 80
standalone
 converting clients to, 83-86
SWAPSIZE variable, 47
swap space
 changing the size of, 76
 on local disks, 76, 79
swap trees
 configuring, 21
 contents of, 6, 19
 creating, 21
 installing, 56
 planning, 19
 removing manually, 90
 root directory of, 14
 space requirements for, 19
 when to build, 22
SWAP variable, 47
system disks, 76, 78

T

technical assistance
 calling, 94
tftp protocol, 3, 9, 91
Trivial File Transfer Protocol. *See tftp* protocol.
troubleshooting
 booting, 91
 client_inst errors, 89
 network problems, 87

U

uname command, 9
updating client trees, 67
updating share trees, 64-66
User Datagram Protocol, 91

/usr/etc/bootptab file, 91

V

verbose NVRAM variable, 93
verifying client operation, 82
VIDEO variable, 33, 51
virtual memory, 6

