

IRIX[®] Admin
Software Installation and Licensing

Document Number 007-1364-110

CONTRIBUTORS

Written by Pam Sogard, with contributions by Susan Ellis and John Raithel

Updated by Bob Bernard

Illustrated by Dany Galgani

Edited by Christina Cary

Production by Linda Rae Sande

Engineering contributions by Ken Chin, Mark Minnihan, David Olson, John Relph,
Will Rusch, and Beth Styles

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

© Copyright 1996 - 1999 Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole
or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by
the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the
Rights in Technical Data and Computer Software clause at DFARS 52.227-7013
and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR
Supplement. Unpublished rights reserved under the Copyright Laws of the United
States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd.,
Mountain View, CA 94043-1389.

Indy, IRIX, and Silicon Graphics are registered trademarks and IRIS InSight, O2,
RoboInst, the Silicon Graphics logo, and XFS are trademarks of Silicon Graphics, Inc.
Alias/Wavefront is a trademark of Alias/Wavefront, a division of Silicon Graphics
Limited. FLEXlm is a trademark of GLOBETrotter Software, Inc. Network License
System and NetLS are trademarks of Apollo Computer, Inc., a subsidiary of
Hewlett-Packard Company. POSIX is a registered trademark of the Institute of
Electrical and Electronic Engineers, Inc. NFS is a registered trademark of Sun
Microsystems, Inc. UNIX is a registered trademark in the United States and other
countries, licensed exclusively through X/Open Company, Ltd.

IRIX® Admin: Software Installation and Licensing
Document Number 007-1364-110

What's New in This Document

New Features Documented

This revision of *IRIX Admin: Software Installation and Licensing* documents an updated user interface for Inst, the Silicon Graphics software installation tool.

Major Documentation Changes

Changes to this document for the IRIX 6.5.3 release include the following:

- Inst interface simplified to install or switch between the maintenance and feature streams of a software release
- updated RoboInst installation procedure to reflect changes in RoboInst version 1.1 from version 1.0

Record of Revision

<i>Version</i>	<i>Description</i>
110	February 1999. Incorporates information for the IRIX 6.5.3 release.

Contents

What's New in This Document	iii
New Features Documented	iii
Major Documentation Changes	iii
Record of Revision	v
List of Examples	xix
List of Figures	xxi
List of Tables	xxiii
IRIX Admin Manual Set	xxv
About This Guide	xxvii
Using This Guide	xxvii
Inst or Software Manager?	xxviii
Installing Diskless Systems	xxviii
What This Guide Contains	xxix
Part I	xxix
Part II	xxx
Appendixes	xxx
Audience for This Guide	xxxi
Notation Conventions	xxxi
Related Documentation	xxxii

- 1. Understanding Software Installations 3**
 - Software Release Types 3
 - Anatomy of a Software Product 4
 - Product Descriptions 5
 - The Installation Database 5
 - Images 5
 - Subsystems 7
 - Required and Default Subsystems 7
 - Preliminary Selections 7
 - Factory-Installed Subsystems 8
 - Sources of Installable Software 8
 - Software Installation Methods 10
 - Introduction to Inst 10
 - Automatic Initialization 11
 - The Main Menu 11
 - Changing the Distribution Source 12
 - Viewing the Software Inventory 12
 - Selecting and Installing Software 12
 - Handling Conflicts 13
 - Ending a Session 13
 - Supplementary Menus 13
 - Getting Help During Installations 14
- 2. Preparing for Installation 15**
 - Selecting a Distribution Source 15
 - Setting Up an Installation Server 16
 - Enabling BOOTP Forwarding on Routers 16
 - Enabling TFTP Access on an Installation Server 17
 - Configuring an Installation Account 18
 - Creating Distribution Directories 20
 - Backing Up the Target Systems 22
 - Opening Multiple Distributions 23

- Collecting the Information That You Need 23
 - Getting CD-ROM Device Numbers 23
 - Getting the Target's Name and Address 24
 - Getting the Server's Name and Address 24
 - Getting the Installation Account Name 25
- 3. Starting an Installation 27**
 - Determining the Installation Method 27
 - Preparations Checklist 28
 - About Miniroot Installations 29
 - Starting a Miniroot Installation 30
 - Loading From a Local CD 30
 - Loading From a Remote Distribution Source 34
 - Starting a Live Installation 36
- 4. Using the Main Menu 37**
 - The Software Installation Procedure 38
 - Step 1: Invoking Inst 40
 - Step 2: Specifying the Source 41
 - Using the from Command 41
 - Using the open Command 42
 - Automatic Initialization Functions 42
 - Step 3: Displaying Software Information 43
 - Using the list Command 43
 - Displaying Software Installed on the Target 44
 - Interpreting list Output 45
 - When Multiple Distributions Are Open 47
 - Interpreting the Disk Space Summary 47
 - Step 4: Making Software Selections 48
 - Using the install, remove, and keep Commands 48
 - Simple Selection Command Entries 49
 - Using Arguments in Selection Commands 50
 - Checking Your Selections With list 50
 - Accepting Automatic Selections 50

- Step 5: Launching the Installation 51
- Step 6: Resolving Conflicts 52
 - Using the conflicts Command 53
 - Resolving Other Errors 54
 - Errors Caused by Read-Only Directories 54
 - Errors that Display the Interrupt Menu 54
- Step 7: Quitting the Installation Session 55
 - If Selections Are Pending 56
 - If Conflicts Are Unresolved 56
 - If Rebooting Is Needed 57
- If Inst Is Interrupted 57
 - Completing a Checkpoint Restart 57
 - If a Checkpoint Restart Fails 58
- Managing Configuration Files 60
 - Configuration File Anomalies 60
 - The Configuration File Notice 61
 - Merging Configuration Files 61
- 5. Using Supplementary Menus 63**
 - Giving Commands on Other Menus 63
 - Using the View Commands Menu 64
 - The target and distribution Commands 65
 - The filter Command 66
 - The level Command 67
 - The sort Command 68
 - The sizes and deltas Commands 68
 - The filesystems Command 69

Using the Administrative Commands Menu	70
The set Command	71
The date Command	72
The files Command	72
The space Command	74
The recalculate Command	74
The sh and shroot Commands	75
The relocate Command	75
The mount and umount Commands	76
The mkfs Command	76
The sethostname and sethostipaddr Commands	77
The save and load Commands	77
The config Command	78
The hardware Command	79
The updatekeepfile Command	79
Using the Interrupt Menu	79
Retrying and Operation	80
Stopping an Installation	80
Using Other Interrupt Menu Commands	81
6. Inst for Experts	83
Alternative Ways to Specify a Distribution	83
Using the \$dist Variable	83
Specifying One Product as the Source	84
Specifying an Alternate User Account	84
Specifying the Source on the Command Line	84
Installing Software Noninteractively	85
Using Command Shortcuts	85
Giving Commands by Number	85
Abbreviating Command Names	86
Using Wildcards in Subsystem Names	86
Using Product and Image Names	87
Using Keywords as Arguments	87
Using a Distribution Keyword	90

- Using step to Make Selections 90
- Using Advanced step Features 91
- 7. Maintenance Tips 95**
 - Installing Maintenance and Feature Releases 95
 - Switching Streams 96
 - Feature-to-Maintenance Switches 96
 - Maintenance-to-Feature Switches 96
 - Maintenance and Feature Information in List Displays 97
 - Maintenance and Feature Release Installation Procedure 97
 - Installing Optional Software Products 99
 - Installing Patches 99
 - Interpreting Patch Information in list Displays 100
 - Patch Installation Procedure 101
 - Removing Patches 102
 - Removing Original Product Files 102
 - Installing Software for Hardware Upgrades 103
 - Installing Accompanying Product Releases 103
 - Reinstalling the Same Software 103
 - Installing Reference Pages 104
 - Reinstalling an Older Software Release 105
 - Avoiding Compatibility Problems 108
 - Removing Software 109
- 8. Tailoring Installations 111**
 - Using an Initialization Script 112
 - Syntax of the inst.init Script 112
 - Writing an inst.init Script 113
 - Additional inst.init Script Rules 114
 - Sample inst.init Scripts 114
 - Using README Files in a Distribution 115
 - Using a Related Distribution File 116

Setting Up an HTTP Installation Server	116
Using a Tardist Installation File	117
Using an HTML Distribution Directory	118
Checking the Current Preference Settings	119
Controlling Menu Displays	120
Turning Off Menu Displays	120
Displaying Hidden Commands	120
Redisplaying the Menu	121
Controlling Disk Space Checks and Reports	121
Estimated and Detailed Space Checks	121
Changing Detailed Space Checking	122
Delaying Disk Space Checking	122
Minimizing Disk Space-Checking Time	123
Controlling the Display of Filenames	123
Displaying the Current Distribution Source	124
Turning Off Initial Selection of Subsystems	125
Changing the Network Time-out Value	125
A Closer Look at Preferences	126
The Preferences File	126
Creating Additional Preferences Files	127
The Preferences List	128
9. Automating Installations With RoboInst	147
How RoboInst Works	147
Events in a RoboInst Installation	150
Disk Partitions and RoboInst	150
Getting Ready for RoboInst	151
The Client System List	151
Software Distribution Server Setup	151
Boot Server Setup	152
Configuration Server Setup	152
Configuration Directory Setup	153
Network Setup	153

- Customizing Miniroot Installations 154
 - Creating a Miniroot Configuration File 154
 - Ordering Commands in an mrconfig File 156
 - Partition Keyword 157
 - Hostname Considerations 158
 - Environment Variables 159
 - Creating Additional Configuration Files 160
 - Example Boot and Configuration Server 160
- Launching RoboInst 162
 - Launching RoboInst From the Server 162
 - Launching RoboInst from the Client 163
 - Launching RoboInst From the PROM 163
 - Additional RoboInst Command-Line Options 165
- 10. Licensing 169**
 - About License Manager 169
 - License Types 170
 - The License Request Process Illustrated 171
 - FLEXlm Files and Utilities on Silicon Graphics Systems 172
 - The License File Database for Silicon Graphics Applications 173
 - Specifying License File Locations 174
 - New FLEXlm Features 174
 - The USE_SERVER Feature 174
 - The port@host Feature 175
 - Starting and Stopping Licensing Daemons 176
- A. Inst Quick Reference 177**
 - Condensed Installation Instructions 177
 - Main Menu Commands 180
 - View Menu Commands 187
 - Administrative Menu Commands 189
 - Interrupt Menu Commands 192

B.	Troubleshooting Installations	195
	Types of Errors	196
	Fatal	196
	Error	196
	Warning	196
	Resolving Errors	197
	Errors Loading the Miniroot	197
	Errors Loading the Miniroot From Local CD	198
	Errors Loading the Miniroot From Remote Host	199
	Other Errors Loading the Miniroot From a CD	204
	Questionable Miniroot Image: IRIX 6.3 and Earlier	204
	Questionable Miniroot Image: IRIX 6.4	205
	Errors While Starting an Installation Session	207
	Wrong Diskless Modes	208
	Errors Starting Live Installations	209
	Inst Library libinst.so Errors	210
	Errors in the Distribution	210
	Preinstallation Check Errors	212
	Errors While Installing and Removing Software	214
	Disk Space Errors	215
	Subcommand and Exitop Errors	217
	Network Time-out Errors	218
	Archive Corrupt Errors	219
	Device Busy Errors	219
	Errors Leaving an Installation Session (RQS Errors)	220

- Resolving Conflicts 220
 - Resolving Different Types of Conflicts 221
 - Preexisting Conflicts 221
 - Required Product Conflict 222
 - Prerequisite Conflict 223
 - Back Prerequisite Conflict 223
 - Incompatible Products Conflict 223
 - Downgrade Conflict 224
 - Overriding Conflicts 224
- Resolving Network Problems 225
 - Checking Network Connections 225
 - Network Problem Diagnosis During an Inst Session 227
 - Network Problem Diagnosis During Miniroot Installations 229
- Resolving Problems With CDs 231
 - Checking CD-ROM Drives 231
 - Verifying That a CD-ROM Drive Is Recognized 232
 - Verifying That a CD Is Mounted 233
 - Checking Distribution Directories and CDs 234
 - Using fx to Restore the Swap Partition 235
 - Calling Your Support Organization 236
- Index 239**

List of Examples

Example 4-1	The open Command	42
Example 4-2	The list Legend	43
Example 4-3	Distribution Software Posted by the list Command	44
Example 4-4	Target Software Posted by the list Command	45
Example 4-5	List Output for Multiple Open Distributions	47
Example 4-6	Disk Space Summary	47
Example 4-7	Multiple Arguments to the install, remove, and keep Commands	50
Example 4-8	List of Subsystems Selected for Installation	51
Example 4-9	Successful Installation Messages	52
Example 4-10	Conflicts Message	53
Example 4-11	Entry to Resolve Conflicts	53
Example 4-12	Error With Directory Permissions	54
Example 4-13	Error in Installation Processing	55
Example 4-14	Conflicts Message When Quitting Inst	56
Example 4-15	Quit Prompt With Reboot Message	57
Example 4-16	Notice of Configuration File Changes	61
Example 4-17	Listing of Changed Configuration Files	61
Example 5-1	A list Display From the Target View	65
Example 5-2	A step Display From the Distribution View	66
Example 5-3	Effects of Using View Filters	67
Example 5-4	Effects of level on a list Display	67
Example 5-5	Effects of sort on a list Display	68
Example 5-6	Displaying Individual Filesystems	69
Example 5-7	Changing Column Order in list Displays	70
Example 5-8	Sample files Display	72
Example 5-9	Sample Selections File	78
Example 6-1	Sample of a Noninteractive Installation	85

Example 6-2	Commands With Product and Image Arguments	87
Example 6-3	Sample Product, Image, and Subsystem Names	87
Example 6-4	A step Command Display	91
Example 7-1	Sample Listing for Maintenance or Feature Software	97
Example 7-2	Sample list Display of Patches	100
Example 8-1	Sample inst.init Script	114
Example 8-2	Sample inst.init Script	115
Example 8-3	HTTP Selections File for a Tardist File Installation	117
Example 8-4	HTML Installation File	119
Example 8-5	Sample <i>.swmgrrc</i> File	126
Example 9-1	RoboInst Configuration Test	162
Example 9-2	RoboInst Configuration Launch With Prompting	162
Example 9-3	RoboInst Installation From the Client	163
Example 10-1	Database of License Locations	173
Example 10-2	Client License File With the USE_SERVER Feature	175
Example 10-3	Specification for the Enhanced port@host Feature	175
Example 10-4	Specification for Unenhanced Processing of port@host	175

List of Figures

Figure 1-1	Components of a Software Release	6
Figure 1-2	Software Distribution Sources	9
Figure 1-3	The Main Menu for Inst	11
Figure 3-1	System Shutdown Messages	31
Figure 3-2	System Restart Message	31
Figure 3-3	The System Maintenance Menu	32
Figure 4-1	Summary of the Installation Procedure	39
Figure 4-2	The Inst Main Menu	40
Figure 4-3	The Checkpoint Restart Menu	57
Figure 5-1	The View Menu	64
Figure 5-2	The Administrative Commands Menu	71
Figure 5-3	The Interrupt Menu	80
Figure 9-1	Simple RoboInst Configuration	149
Figure 9-2	Sample RoboInst Directory and File Setup	161
Figure 10-1	License Manager Dialog	170
Figure 10-2	License Request Process	171
Figure B-1	The Interrupt Menu	215

List of Tables

Table 6-1	Subsystem Names Specified With Wildcard	86
Table 6-2	Keywords	88
Table 6-3	Requests in <i>step</i> Operations	92
Table 8-1	Preferences That Affect Inst and Software Manager	128
Table 8-2	Preferences That Affect Inst Only	138
Table 8-3	Preferences That Affect Software Manager Only	142
Table 9-1	Keywords in an mrconfig File	154
Table A-1	Help Topics	182
Table A-2	Keys Used in step Mode	185

IRIX Admin Manual Set



This guide is part of the *IRIX Admin* manual set, which is intended for administrators: those who are responsible for servers, multiple systems, and file structures outside the user's home directory and immediate working directories. If you maintain systems for others or if you require more information about IRIX than is in the end-user manuals, these guides are for you. The *IRIX Admin* guides are available through the IRIS InSight online viewing system. The set comprises these volumes:

- *IRIX Admin: Software Installation and Licensing* (this document)—Explains how to install and license software that runs under IRIX, the Silicon Graphics implementation of the UNIX operating system. Contains instructions for performing miniroot and live installations using Inst, the command-line interface to the IRIX installation utility. Identifies the licensing products that control access to restricted applications running under IRIX and refers readers to licensing product documentation.
- *IRIX Admin: System Configuration and Operation*—Lists good general system administration practices and describes system administration tasks, including configuring the operating system; managing user accounts, user processes, and disk resources; interacting with the system while in the PROM monitor; and tuning system performance.
- *IRIX Admin: Disks and Filesystems*—Explains disk, filesystem, and logical volume concepts. Provides system administration procedures for SCSI disks, XFS and EFS filesystems, XLV logical volumes, and guaranteed-rate I/O.
- *IRIX Admin: Networking and Mail*—Describes how to plan, set up, use, and maintain the networking and mail systems, including discussions of sendmail, UUCP, SLIP, and PPP.
- *IRIX Admin: Backup, Security, and Accounting*—Describes how to back up and restore files, how to protect your system's and network's security, and how to track system usage on a per-user basis.
- *IRIX Admin: Peripheral Devices*—Describes how to set up and maintain the software for peripheral devices such as terminals, modems, printers, and CD-ROM and tape drives.
- *IRIX Admin: Selected Reference Pages* (not available in InSight)—Provides concise reference page (manual page) information on the use of commands that may be needed while the system is down. Generally, each reference page covers one command, although some reference pages cover several closely related commands. Reference pages are available online through the man(1) command.

About This Guide

IRIX Admin: Software Installation and Licensing explains how to install and license software that runs under IRIX, the Silicon Graphics implementation of the UNIX operating system. The IRIX installation utility, *inst*, supports two user interfaces: *Inst*, a command line interface; and Software Manager, a graphical user interface. IRIX installation tools also include RoboInst, optional software that performs multiple installations on remote systems simultaneously.

This guide contains three parts: Part I and Appendix A cover installing with *Inst*, the command line interface; Part II covers installing with RoboInst; and Part III covers licensing.

Note: Software Manager is documented in the *Personal System Administration Guide*.

Using This Guide

IRIX Admin: Software Installation and Licensing explains how to install Silicon Graphics software products using *Inst*, the command-line interface to the IRIX installation utility, *inst* (see the *inst(1M)* reference page). It introduces readers to Silicon Graphics software release conventions and explains how to plan, execute, and maintain a software installation on Silicon Graphics workstations and servers.

Two installation methods are described in this guide:

- *live installations*, in which software is transferred while the target system remains in operation
- specialized installations, called *miniroot installations*, that rely on system firmware for the software transfer (this method requires a system shutdown).

This guide also contains information that is specific to setting up *FLEXlm* licensing software on Silicon Graphics workstations and servers. Please refer to the following documents, which are available online from IRIS InSight, for complete information on administering a licensing scheme:

- *FLEXlm End User Manual*
- *Network License System Administration Guide*

Inst or Software Manager?

If you have IRIX 6.3 (or later) installed on the target system, you can use either Inst or Software Manager to perform live or miniroot installations. The tool that you choose depends on whether you prefer using a command-line or graphical interface. If unusual errors occur during a miniroot installation, the Inst interface is automatically invoked for error recovery (even if you used Software Manager to start the installation), since Inst is the only interface available from the miniroot (“About Miniroot Installations” in Chapter 3 provides more information).

Note: If the target system is running IRIX 6.2 or earlier, you must use Inst to perform any miniroot installation, including an upgrade to IRIX 6.5.

The operation of *inst* and the Inst and Software Manager interfaces can be customized with a set of preferences. Use Chapter 8 of this guide as a comprehensive reference for specifying preferences and for managing the configuration files that store preference settings for both interfaces.

Installing Diskless Systems

IRIX Admin: Software Installation and Licensing does not explain how to install software for diskless operation. Please refer to the *Diskless Workstation Administration Guide* for instructions on installing diskless systems.

What This Guide Contains

The *IRIX Admin: Software Installation and Licensing* is organized into two parts and two appendixes:

Part I

Part I of this guide consists of eight chapters covering these installation topics:

Chapter 1, “Understanding Software Installations”

Describes how software distributions are organized and the effects of this organization on the installation process. Introduces the fundamental concepts and vocabulary of software installation and presents an overview of the installation process using Inst.

Chapter 2, “Preparing for Installation”

Identifies the factors that must be considered in planning for installations and describes the preliminary work on a server and target system that ensures smooth installations.

Chapter 3, “Starting an Installation”

Contains procedures for starting a miniroot and a live installation. Also explains how miniroot installations work. Use this chapter in conjunction with the procedures in Chapter 4 to install software.

Chapter 4, “Using the Main Menu”

Contains procedures for installing software using commands on the Inst Main menu. Use this chapter after starting an installation with the procedures in Chapter 3.

Chapter 5, “Using Supplementary Menus”

Describes how to use the View Commands menu to change the way information is displayed in software listings. Also explains how to use the Administrative Commands menu to execute specialized installation commands and change conditions on the target host during an installation session. Procedures in this chapter require mastery of the material in Chapter 4.

Chapter 6, “Inst for Experts”

Describes methods of using Inst that supplement the basic method presented in Chapter 4. Use this chapter after you become comfortable with the procedures in Chapter 4.

Chapter 7, “Maintenance Tips”

Contains procedures for updating installed software and keeping software current with site requirements.

Chapter 8, “Tailoring Installations”

Explains how to customize Inst to suit both personal preferences and the requirements of other system users. Includes information for tailoring Software Manager.

Chapter 9, “Automating Installations With RoboInst”

Explains how automatic miniroot installations work, how to prepare the files that they need, and how to launch automatic miniroot installations on multiple target systems.

Part II

Part II of this guide contains one chapter:

Chapter 10, “Licensing”

Refers readers to the *FLEXlm End User Manual* and the *Network License System Administration Guide* for instructions on setting up and maintain licensing schemes. Contains supplemental information that is specific to managing FLEXlm on Silicon Graphics systems.

Appendixes

Two appendixes are provided:

Appendix A, “Inst Quick Reference”

Contains a condensed version of the installation instructions presented in Chapter 4. Also contains a glossary of all commands that are available from Inst menus, including hidden commands. Commands are ordered alphabetically by menu.

Appendix B, “Troubleshooting Installations”

Explains the types of errors that can occur during an installation, how to control error reporting, and how to interpret and resolve errors during various phases of the installation process.

Audience for This Guide

The audience for the *IRIX Admin: Software Installation and Licensing* is experienced UNIX system and site administrators who are responsible for putting software into service on various models of Silicon Graphics workstations and servers. It may also be used by individual workstation owners as a comprehensive guide to software installation.

Procedures in this guide require these proficiencies:

- for routine installations, experience with a command-line interface and familiarity with UNIX documentation conventions
- for site preparation procedures, competence with managing servers and communications between network nodes
- for advanced troubleshooting and installing on systems with nonstandard filesystem configurations or multiple system disks, proficiency with UNIX filesystem and disk management

Notation Conventions

This document uses standard UNIX conventions when referring to IRIX reference pages. A command name is followed by a section number in parentheses. For example, `mkfs(1M)` refers to the *mkfs* command entry in Section 1M of the IRIX reference pages.

In text, descriptions appear in italics. Screen text, command syntax descriptions, and user entries follow these conventions:

`typewriter font`

Menus, prompts, error messages, and other screen displays

typewriter bold

Commands and literal arguments, keywords and their abbreviations, and any text that is entered exactly as shown

italics

Filenames, directories, commands and command arguments, variables, and any other portion of an entry for which the user must supply a value.

<>

(angle brackets) Optional components in an entry

Related Documentation

Refer to the documents listed below for supplementary information.

- *CRAY Origin2000: Upgrading an IRIX Operating System on a /target_root* for IRIX operating system upgrade procedures for large (64+ processors) CRAY Origin2000 systems.
- *IRIX Admin: Disks and Filesystems* for information on IRIX filesystem management
- *IRIX Admin: Networking and Mail* for information on network administration
- The owner's guide to target systems for specific information on using the services of the programmable read-only memory (PROM)
- *Personal System Administration Guide* for instructions on using the Software Manager
- *Diskless Workstation Administration Guide* for information on installing software for diskless clients
- *FLEXlm End User Manual* and *Network License System Administration Guide* for information on administering software licenses

PART ONE

Installation

Chapter 1, “Understanding Software Installations”

Chapter 2, “Preparing for Installation”

Chapter 3, “Starting an Installation”

Chapter 4, “Using the Main Menu”

Chapter 5, “Using Supplementary Menus”

Chapter 6, “Inst for Experts”

Chapter 7, “Maintenance Tips”

Chapter 8, “Tailoring Installations”

Understanding Software Installations

Silicon Graphics software is distributed on compact discs (CDs). A CD contains one or more software products and any special tools that the products require for installation. The purpose of the installation utility, *inst*, is to transfer software in the distribution, which has an encoded format, to a hard disk in a usable format. The installation utility offers two user interfaces: a graphical interface called Software Manager, documented in the *Personal System Administration Guide*; and a command-line interface, called *Inst* the subject of this guide.

A distribution CD always includes installable release notes that provide installation instructions, including any prerequisites and incompatibilities information about the software that the CD contains. You can read product release notes from the distribution CD using the *CDgrelnotes* or *CDrelnotes* command. Instructions for reading release notes are included in the CD jewel case insert.

Software Release Types

Any software distribution is one of several release types; the type is determined by the purpose of the software in the release. A *base release* of the IRIX operating system upgrades the fundamental services that support all hardware and software functions. When a base release is installed, all components of the operating system are replaced by a new version. The new base software operates on all workstation and server models, unless the release specifically designates an older model for obsolescence.

Intermediate releases are distributed subsequent to a base release. They correct flaws in the base release or introduce additional hardware and software features to it. Because the new files in an intermediate release overwrite the existing files that they are replacing, they are also referred to as *overlays*.

A base release and the intermediate releases that apply to it constitute a *release family*. The following intermediate releases may be members of the family:

Maintenance release

contains device drivers for new hardware, replacement software to correct flaws in a base release, or both.

Feature release

contains the software in any maintenance release that precedes it, as well as new software options and features that enhance the base release but are not required for its proper functioning.

Patches

contain a collection of one or more files that correct a serious and urgent problem in a software release, such as a threat to system security or data integrity. Patches are issued when a remedy is required before the next scheduled maintenance release. Patches are also issued to correct a problem that occurs only on a particular hardware platform.

A patch can apply to a base release, a maintenance release, or a feature release. When you install a patch, the existing files are moved to a holding directory and the new files are installed in their place.

Only maintenance releases or feature releases are installed onto a particular base release. For example, if you choose to install a maintenance release on the IRIX 6.5 base release, all subsequent installations will be maintenance releases (or patches to a maintenance release) until you install the next base release.

Anatomy of a Software Product

A *software product* is a collection of files that support either an operating system function or a specific application. A small number of products support critical functions and must be installed if the system is to operate; other products are not critical but optimize system operation and are recommended for installation by the manufacturer. A subset of required and recommended products is installed in servers and workstations before they leave the factory.

Products are known to Inst by a *shortname*, which is an abbreviation of the formal product name. For example, the shortname of the Fortran 77 Compiler product is *ftn77_dev*. When you install a product, files from the previous version are automatically removed before the new files are installed (see “Managing Configuration Files” on page 60 for exceptions). If a release includes hardware-specific files, the installation utility automatically determines the correct version for a particular model and installs it.

A software product contains three elements: a *product description*, an *installation database*, and one or more *images*, which are composed of *subsystems*. Figure 1-1 illustrates a distribution containing four software product releases.

Product Descriptions

The *product description* is a file that contains information about product requirements and the installation environment for the product. The name of the product description file is the shortname for the product. For example, the name of the product description file for the Fortran 77 Compiler product is *ftn_dev*. Because it contains product specifications, the product description is also referred to as the *spec file*.

The Installation Database

The *installation database* is a file that contains installation information for every file in the product. The name of the installation database is the shortname with an *.idb* extension. For example, the name of the installation database for the Fortran 77 Compiler product is *ftn_dev.idb*. The installation database for a product is also referred to as the *idb file*.

Images

An *image* is a collection of installable files that perform a similar or complementary function. Typically, each software product contains at least two images. This modularity allows for tailoring installations. For example, it is possible to install the images that contain the executable programs of a product without installing the release notes image.

The name of an image is the product shortname followed by an extension that describes the image's function. For example, *ftn_dev.sw* is the image in the Fortran 77 Compiler product that contains files for program execution, and *ftn_dev.man* is the image that contains documentation files.

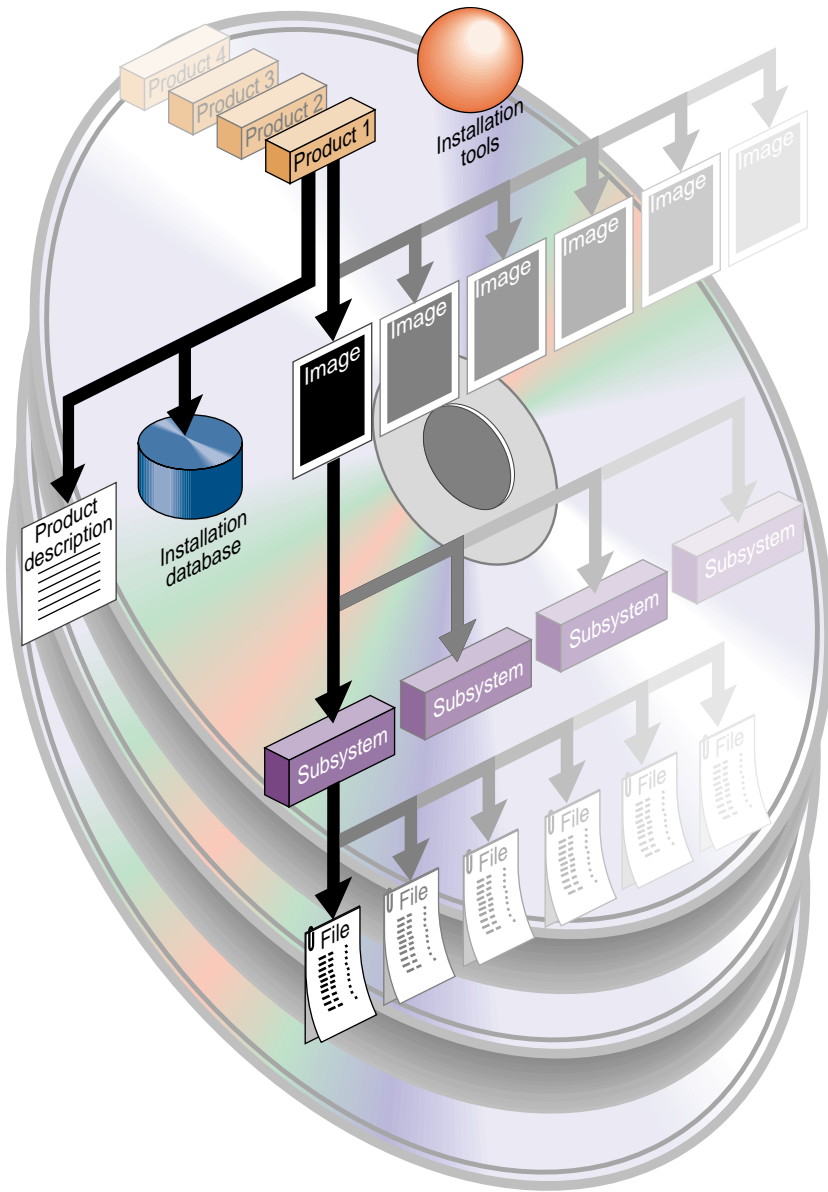


Figure 1-1 Components of a Software Release

Subsystems

Images are composed of one or more *subsystems*. In a software product release, a subsystem is the smallest installable unit of software; the files that make up a subsystem cannot be installed individually (see “Sources of Installable Software” on page 8 for exceptions). Subsystems are assigned three-part names in this form:

product.image_type.subsystem_descriptor

For example, *ftn.sw.ftn* and *ftn.sw.utils* are subsystems in the Fortran 77 Compiler product.

Required and Default Subsystems

Some subsystems in a software product are classified as *default* subsystems. Default subsystems support the basic functions of a product and are suggested for installation by the manufacturer. When disk space is limited, the default subsystems serve as a guideline for determining what to install.

Some default subsystems are also classified as *required* subsystems. Required subsystems are critical to system operation; without them, IRIX will not operate. Inst prevents users from exiting an installation session if a required subsystem is available for installation but not installed (unless you choose to remove this safeguard).

Preliminary Selections

When Inst is initialized, it makes preliminary selections of the subsystems to be installed during the session. If a target system contains no software, the preliminary selections always include all required and default subsystems in the distribution. On targets with software installed, the preliminary selections are all upgrades to the installed software and all default subsystems of any new products that the distribution contains. When subsystems are displayed during an installation session, upgrade subsystems are tagged with the letter *U* to the left of the subsystem name; default subsystems are tagged with the letter *d*, and required subsystems are tagged with the letter *r* to the right of the subsystem name.

Factory-Installed Subsystems

Some subsystems from the *eeo*, *motif_eeo*, and other standard (no-cost) products are factory installed. The *eeo* product contains basic operating system software, utilities, and window system software. Extra-cost software options such as compilers are not factory installed.

To display the inventory of software that is installed on a system when you receive it, use the program *showprods* (see the *showprods(1M)* reference page):

```
% showprods
```

The output of *showprods* is one line for every product, image, and subsystem installed on the target. You can remove any factory-installed software that you do not need, except required subsystems.

Sources of Installable Software

The location of a software distribution is known as the *distribution source*; the system receiving software during an installation is known as the *target*. A distribution source may be a CD that is mounted on the target, a CD that is mounted on a remote system, or a centralized directory on the network to which the distribution has been copied (see Figure 1-2). When a distribution CD contains operating system software, the software is stored in a directory called */dist*.

A server or personal workstation that supplies a distribution source to remote targets is known as an *installation server*. An installation server can provide the distribution source from a local CD-ROM drive or from a disk directory that contains the installable images. A directory of installable software is known as a *distribution directory*. A distribution directory may contain software from several distributions.

Figure 1-2 illustrates local and remote distribution sources.

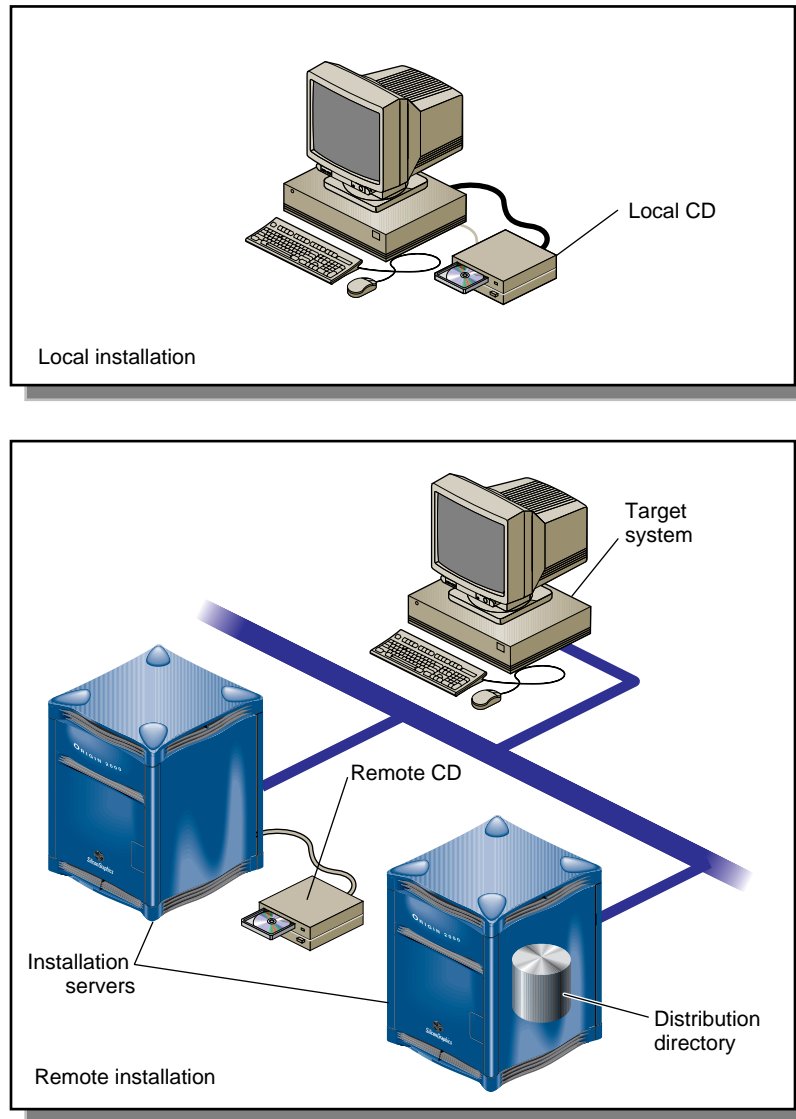


Figure 1-2 Software Distribution Sources

Software Installation Methods

As of IRIX 5.3, most software installations can be performed without interrupting system operations on the target. Installations that are performed without shutting down the system are referred to as *live installations*. Live installations are preferred because they are usually less time consuming and because other system operations can be sustained during the installation session.

Note: Live installations can be performed with both Inst and Software Manager. See Chapter 3, “Starting an Installation,” for more information.

Whenever software installation affects fundamental IRIX functions (such as device management), software must be installed by a *miniroot installation*, which requires a system shutdown. During a miniroot installation, a special set of installation tools is copied to the swap partition of the system disk on the target. The software environment that these tools create for the installation is referred to as the *miniroot*. Only a portion of the IRIX operating system is available in the miniroot (see “About Miniroot Installations” in Chapter 3 for more information).

Note: (Applies to IRIX 6.3 or later) In Software Manager sessions, a miniroot installation is automatically performed if required by any subsystem specified for installation. The system shutdown and restart are transparent to the user except for informational messages, and in most cases, no user interactions are required in the miniroot.

Introduction to Inst

Inst is invoked automatically in a miniroot installation; it is invoked from the command line in a live installation. After Inst is invoked, its Main menu is displayed.

Note: It might be useful to look at Figure 4-1 on page 39 for an illustration of the installation process.

Figure 1-3 illustrates the Main menu.

Inst 3.4 Main Menu

1. from [source ...]	Specify location of software to be installed
2. open [source ...]	Specify additional software locations
3. close [source ...]	Close a software distribution location
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]	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 Menu
13. admin ...	Go to the Administrative Commands Menu
14. quit	Terminate software installation

Inst> █

Figure 1-3 The Main Menu for Inst

Automatic Initialization

During initialization, Inst reads the product descriptions and the installation database, makes preliminary selections of subsystems to install (see “Required and Default Subsystems” on page 7), and calculates estimated space requirements. Initialization is triggered when you issue the first command of the session that requires Inst to collect information about either the distribution or the target software.

The Main Menu

The Main menu contains all commands that are needed to install and remove software. When the Main menu is displayed, Inst posts this prompt to accept command entries:

Inst>

Changing the Distribution Source

Inst always assumes that the installation target is the system on which it is running. Inst selects a default distribution source, which, for live installations, is the distribution source of the previous installation; for miniroot installations, the default source is the location from which the miniroot was invoked (usually the local CD-ROM). You can change a distribution source using the *from* command and specify additional distributions with the *open* command during an installation session.

Viewing the Software Inventory

The *list* command displays an inventory of the software on either the distribution or the target system. Displaying software inventories early in a session is helpful in deciding what software to install or remove. By default, the output of *list* displays the shortnames of subsystems in the distribution inventory, their size and description, and selection and installation information.

Using the *view* command, you can toggle the *list* command so that it displays either the distribution or the target inventory. The *view* command also focuses list output on different levels of the product hierarchy so that you can display information about subsystems, images, or entire products (see Figure 1-1 on page 6). You can also use *view* to filter certain information from *list* displays.

Selecting and Installing Software

You can install the preliminary installation selections that Inst makes during initialization or change these selections before launching an installation. The *install* command selects items in the distribution inventory for installation; the *remove* command selects items on the target inventory for removal. The *keep* command selects items on either list for which no action should be taken. A special selection command, *step*, allows you to list software components individually and select them as each component is listed.

You can change selections as many times as necessary as long as you do not issue the *go* command, which launches the installation. The *go* command initiates the installation and removal of software that you specified in your selections.

Handling Conflicts

Sometimes the selections that you make for installation or removal cause conflicts. For example, a conflict occurs if you select two incompatible subsystems for installation, if you select a subsystem for installation and do not select its prerequisites, or if you close a distribution before installing the products that you selected from it. When conflicts occur, the *go* command does not execute, and Inst posts a conflict advisory that suggests a resolution to the conflict. When you see a conflicts advisory, you can either use the *conflicts* command to accept one of the suggested resolutions or use the *install* and *remove* commands to change your selections.

You also see an advisory if your selections require more disk space than is safely available. This condition requires that you change your selections or remove software from the target system.

Ending a Session

To end an Inst session, enter the *quit* command. When you quit an installation session, Inst performs cleanup operations on target filesystems and on the newly installed software.

Supplementary Menus

Inst offers three supplementary menus that you can use to augment Main menu functions:

- the View Commands menu
- the Administrative Commands menu
- the Interrupt menu

The View Commands menu (shown in Figure 5-1 on page 64) contains selections that let you control information displays during the session. For example, a listing of distribution software normally includes all products that the distribution contains. Using the View Commands menu, you can request a list of the new software products only.

The Administrative Commands menu (shown in Figure 5-2 on page 71) contains two types of selections: commands that let you display or adjust conditions on the target system, such as the *mount* and *umount* commands, and commands that let you control Inst operations, such as the *recalculate* command.

The Interrupt menu (shown in Figure 5-3 on page 80) is used to suspend command processing. It is rarely necessary to display this menu during a session, and Inst displays it only under unusual error conditions.

Getting Help During Installations

During an installation session, you can use the *help* command from any menu to get information about almost every Inst command and command argument. Enter the *help* command alone to get general instructions on using online help. Enter the *help* command followed by an argument to get help on the topic specified in the argument. Table A-1 on page 182 gives a complete list of topics for which help is available.

This sample illustrates a *help* entry and the information that it provides:

```
Inst> help upgrade
```

```
upgrade U
```

```
"upgrade" is a keyword argument for the "install", "keep", "list", and  
"step" commands. For example, "install U" requests that all subsystems  
that are currently installed and have newer counterparts in the  
software distribution be installed...
```

Understanding installation terminology and the Inst installation model is important to using Inst successfully. To get this information, issue this command during an installation session:

```
Inst> help overview
```

Preparing for Installation

This chapter contains procedures and information for preparing yourself and your site for software installation. You should review all the information in this chapter, but you do not need to perform all the procedures described. The procedures that you perform will depend on the type of installation and specific conditions at your site.

The chapter contains these sections:

- “Selecting a Distribution Source” on page 15
- “Setting Up an Installation Server” on page 16
- “Backing Up the Target Systems” on page 22
- “Opening Multiple Distributions” on page 23
- “Collecting the Information That You Need” on page 23

Selecting a Distribution Source

When selecting a distribution source, consider the speed and reliability of your network, the frequency with which installations are performed, and the amount of software that will be installed. If disk space is available and the network is fast and reliable, consider creating a centralized distribution directory on an installation server. A centralized directory is particularly useful if users perform their own installations, or if the server’s CD-ROM drive is unavailable.

When you install software on one or two target systems and the targets contain local CD-ROM drives, a locally-mounted distribution CD is often the most efficient distribution source, particularly if your network is slow and you plan to install a lot of software. For any target that is not equipped with a CD-ROM drive, the distribution source must be a remote CD-ROM drive or distribution directory (see Figure 1-2 on page 9).

Setting Up an Installation Server

You can create an installation server on almost any system in your network that is reliable and has adequate disk resources. The distribution source may be a local CD-ROM drive or a distribution directory.

Any system that you plan to use as an installation server must be accessible from the remote targets. This means that communications between the server and targets must support forwarding of boot files (for miniroot installations) and Trivial File Transfer Protocol (TFTP) file transfers. In addition, the installation server must contain a user account that is available to target systems.

The procedures in this section assume that you plan to set up a distribution directory on an installation server.

Note: If you plan to use a remote CD-ROM drive as the distribution source, complete all procedures in this section except “Creating Distribution Directories” on page 20.

Enabling BOOTP Forwarding on Routers

Inst uses the Internet Bootstrap Protocol (BOOTP) to obtain IP addresses during miniroot installations. For this reason, all routers between an installation server and a remote target must allow *bootp* forwarding (see the *bootp(1M)* reference page). On factory-shipped Silicon Graphics systems, however, *bootp* forwarding is disabled in the file */etc/inetd.conf*.

Note: The *inetd.conf* file is stored in */usr/etc* on systems running pre-5.2 releases of IRIX.

The procedure below describes how to identify routers and change the */etc/inetd.conf* file to enable *bootp* forwarding.

Caution: Silicon Graphics, Inc. can not guarantee that this procedure will work on routers that are not Silicon Graphics systems.

1. Determine the names of routers.

Enter this command from the target system:

```
ping -R -c 1 installation_server
```

Look for an entry that begins with *RR:* in the *ping* output. It shows the route of packets from the target to the installation server and back. Each node listed, except the installation server and target, is a router.

2. Become the superuser on the router listed in the *ping* output.

```
% su
Password:
#
```

3. Change the *bootp* line in */etc/inetd.conf* on each router, as needed.

The default *bootp* line in */etc/inetd.conf* looks like this:

```
bootp dgram udp wait root /usr/etc/bootp bootp
```

Add the *-f* flag to enable *bootp* forwarding. The modified line should look like this:

```
bootp dgram udp wait root /usr/etc/bootp bootp -f
```

4. Put your changes into effect.

```
# killall -v -HUP inetd
```

5. Repeat steps 2 through 4 for additional routers.

When the server is no longer needed for software installation, you can return the */etc/inetd.conf* file on the router to its original state, then enter the *killall* command to put the change into effect (see the *killall(1M)* reference page).

Enabling TFTP Access on an Installation Server

During miniroot installations, Inst uses TFTP to transfer files from remote systems. For this reason, any system acting as the server for miniroot installations must allow *tftp* access. On factory-shipped Silicon Graphics systems, *tftpd(1M)* access is restricted in the */etc/inetd.conf* file to */usr/local/boot* and */usr/etc/boot*. You can place the distribution under one of these directories, for example, */usr/local/boot/dist*, or make another directory accessible by editing the *tftpd* entry in */etc/inetd.conf*.

The procedure below explains how to change the */etc/inetd.conf* to allow *tftp* access.

Note: If your site has strict security requirements, step 2 of this procedure offers an option that limits *tftp* access to the distribution directory.

1. Become the superuser on the installation server.

```
% su
```

```
Password:  
#
```

2. Modify the *tftp* line in */etc/inetd.conf* on the server.

You can modify */etc/inetd.conf* to allow *tftp* access that suits the needs of your site. The default *tftp* entry in */etc/inetd.conf* looks similar to this (your entry may contain additional arguments to the *-s* option):

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/local/boot
```

Take any one of these actions to modify the *tftp* entry:

- Remove the *-s /usr/local/boot*, and any additional directories that are specified by the *-s* option, from the entry to allow *tftp* access to all publicly readable directories. (This is not recommended for sites on the Internet.) The modified line should look like this:

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd
```

- Specify access to the distribution directory only (suggested for secure sites). The modified line should look like this:

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s  
                        /usr/local/boot <distdir>
```

- Specify access to the CD-ROM drive. The modified line should look like this:

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s  
                        /usr/local/boot <CDdir>/dist
```

Note: If an account other than *guest* is used for installation, replace the *guest* entry on the *tftp* line with the alternate user ID.

3. Put your changes into effect.

```
# killall -v -HUP inetd
```

When the remote system is no longer needed for software installation, you can return the */etc/inetd.conf* file to its original state, then enter the *killall* command to put your changes into effect.

Configuring an Installation Account

During an installation, Inst defaults to using the *guest* account on the server to accept a connection from the target systems; *guest* must not be password protected. If the *guest* account on the server is either unavailable or password protected (to limit access to the

target by *rsh*, for example), you must provide an alternate means for accessing the server. You can allow access to the server in any of these ways:

- Remove the password from *guest* while installations are taking place.
- Use an account other than *guest* on the server (the alternate account must not be password protected) and specify the alternate account when you start Inst (see “Specifying the Source on the Command Line” on page 84 for details on invoking Inst with the *-f* flag):

```
inst -f alternate_user@installation_server:distdir
```
- Use a password-protected account on the server for installations and create an *.rhosts* file for the installation account. The installation account must have read permissions on the distribution source.

The *.rhosts* file that you create must contain an entry for each target system that will access the server (see the *hosts.equiv(4)* reference page for information on *.rhosts*). For example, assume that the installation account on the server is *instuser*. The file */usr/people/instuser/.rhosts* on the installation server contains these lines to permit installations on targets called *joesbox* and *lab1*:

```
joesbox.engr.xxx.com root  
lab1.engr.xxx.com root
```

Note: This *.rhosts* file does not grant root permissions on the *installation_server*. It only allows the userid *root* on one of the targets to pull install information from the *installation_server*.

Either of the following commands can be used to install software on *joesbox* or *lab1* (see “Step 2: Specifying the Source” on page 41 and “Specifying the Source on the Command Line” on page 84 for details on these commands):

```
Inst> from instuser@installation_server:path  
# inst -f instuser@installation_server:path
```

Creating Distribution Directories

Follow the procedure below to create a parent directory and subsequent distribution directories on the installation server. The CD-ROM drive from which you copy the distribution software may be either a local or a remote drive. In this procedure, you will create a parent directory for your application or operating system, followed by individual directories for the contents of each CD involved.

Caution: Do not install more than one release of a product under a parent directory.

1. Become the superuser on the server with the CD-ROM drive.

```
% su -  
Password:  
#
```

Note: If the distribution directory is on a different system, become the superuser on that system also.

2. Get the controller and unit numbers for the CD-ROM drive.

```
# hinv
```

Look for a line like this in the output of *hinv* and note numbers listed for *CDROM* unit and *SCSI* controller:

```
CDROM: unit 4 on SCSI controller 0
```

3. Eject the CD that is currently in the drive, if any.

You can omit the argument from this command if there is only one CD-ROM drive on the workstation:

```
# eject /dev/scsi/scctlrdunit10
```

Note: This command has a lowercase *l* (not the number 1) and a zero at the end of it.

4. Create the parent directory for the distribution:

```
# mkdir parentdir
```

5. Create a directory for each CD-ROM included with your installation media:

```
# cd parentdir  
# mkdir cd1_directory  
# mkdir cd2_directory
```

... and so on for as many CDs as you have.

6. Stop the CD-ROM daemon, if it is running.

Enter this command on the system with the CD-ROM drive to stop the mediad daemon (see the mediad(1M) reference page):

```
# mediad -k
```

7. Create a mount-point directory for the CD-ROM drive, if none exists.

```
# mkdir /CDROM
```

8. Insert the CD containing the distribution and mount it.

```
# mount -o ro /dev/dsk/dksctlrdunits7 /CDROM
```

9. Copy CD-1's distribution to its directory (created in step 5, above).

On the system containing the distribution directory, copy the installable software from the CD to the distribution directory as illustrated in the following sample commands. Use the `cp` command if the CD-ROM is local, and the `rcp` command if the CD-ROM is remote:

```
# cp -r /CDROM/dist/ /path/to/parentdir/cdx_dir
# rcp -r guest@server:/CDROM/dist/ /path/to/parentdir/cdx_dir
```

You can create a software distribution directory that contains fewer products than are in the CD-ROM distribution by copying the files for just the products that you want. Remember that distribution directories and CD-ROM distributions have an identical structure. For example, to copy just one product from a local CD-ROM drive, use this command:

```
% cp /CDROM/product* product_dir
```

10. Copy the release notes to the server (optional).

Enter this command to create a directory for the release notes and copy them to the installation server:

```
# cp -r /CDROM/relnotes/* relnotes_dir
```

11. Eject the distribution CD.

Enter this command on the system with the CD-ROM drive. You can omit the argument if there is just one CD-ROM drive on the system:

```
# eject /dev/scsi/scntrlrdunit10
```

12. Repeat steps 7 through 9 for additional CDs.

- Restart the CD-ROM daemon, if you stopped it.

If you stopped *mediad* in step 6, restart it by entering this command on the system with the CD-ROM drive:

```
# mediad
```

Note: If you have more than one CD-ROM drive, you might get an error from this command, but you can safely ignore it.

Backing Up the Target Systems

Although backing up the target is not a requirement for installation, it is strongly recommended. You can use several utilities to make backups: System Manager, *backup*, *bru*, *cpio*, and *tar*. See *IRIX Admin: Backup, Security, and Accounting* for detailed information on system backups.

Consider backing up these files:

- User files* that are created or copied to the target. Any file on the target that was not put there during the software installation process is considered a user file.
- Configuration files* that contain information unique to the target system or the site. These files are created during installation but are likely to be modified after they are installed. The unique information in these files is not destroyed during an installation. However, the preinstallation copy of these configuration files is helpful if you decide to go back to the earlier software release after installation. You can identify modified configuration files with this command:

```
# showfiles -c -m -s
```

Use this command to back up the configuration files onto tape:

```
# showfiles -c -m -s | tar cv -
```

Use this command to retrieve specified configuration files from tape and overwrite existing files:

```
# tar xv files
```

Note: If you follow these instructions, you will overwrite the installed configuration file. Refer to “Managing Configuration Files” on page 60 for more information on how configuration files are handled during installations.

Use this command to retrieve all configuration files from tape and overwrite existing files:

```
# tar xv
```

Opening Multiple Distributions

Inst automatically manages the installation order of the products on CDs or in distribution directories. If you expect to install software from more than one CD or distribution directory, you can open each of them with the *open* command. If you attempt to install software that has a pre-requisite of software on an unopened distribution, Inst will prompt you to either open that distribution or not install the software.

Note that if you use the *from* command instead of *open*, any open distributions are closed and any selections lost before the new distribution is opened.

Collecting the Information That You Need

Before starting a miniroot installation, spend a few minutes collecting information that might be required during the installation procedure. Jot down the information that you get as you follow the instructions in this section and have it available when you begin the installation. Remember, it might be impossible to access this information after the installation is under way—the miniroot severely restricts access to files on the target and to network systems that are ordinarily accessible.

Getting CD-ROM Device Numbers

If you are planning a miniroot installation from the local CD-ROM drive on the target, you must be prepared to specify the controller and unit number of the drive during the installation. Use this command to determine the controller and unit number:

```
# hinv
```

Look for a line like the one below in the output of *hinv*. The numbers that you need appear in parentheses beside the *scsi* and *cdrom* fields:

```
SCSI CDROM: scsi (0) cdrom (4)
```

In the previous example, the controller number is 0 and the unit number is 4.

Getting the Target's Name and Address

The name and address of the target system might be required for error recovery during miniroot installations when the distribution source is on a remote installation server.

Enter the *hostname* command from the target system to get its name:

```
% hostname
```

Enter the *ping* command from the target to get its Internet Protocol (IP) address. Use the output of the *hostname* command as an argument to *ping*:

```
% /usr/etc/ping -c 1 targetname
PING targetname (IPaddress): 56 data bytes
----targetname PING Statistics----
1 packet transmitted, 1 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/1/1 ms
```

Getting the Server's Name and Address

The name of the remote installation server is specified to identify the distribution source in miniroot and live installations. If you are working at the installation server, enter the *hostname* command to get its name:

```
% hostname
```

If you are on the target system, enter this *ping* command to get the IP address of the installation server:

```
% /usr/etc/ping -c 1 installation_server
PING installation_server (IPaddress): 56 data bytes
----installation_server PING Statistics----
1 packet transmitted, 1 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 2/2/4 ms
```

Note the numbers in the *IPaddress* field of the output. If *ping* output is not similar to this example, the network connection might be faulty. You should resolve the problem before continuing with your installation (see Appendix B, "Troubleshooting Installations," for help).

Getting the Installation Account Name

By default, Inst uses the *guest* account on the installation server for installations. If you plan to use an account other than *guest*, be prepared to specify the name of the alternate account (see “Configuring an Installation Account” on page 18).

Starting an Installation

One of two procedures is used to invoke Inst for an installation session, depending on whether you plan a miniroot installation or a live installation. This chapter explains how to determine what type of installation is required and provides instructions for invoking Inst for each installation method. It also explains what you must do to prepare for the session before invoking Inst.

This chapter contains these sections:

- “Determining the Installation Method” on page 27
- “Preparations Checklist” on page 28
- “About Miniroot Installations” on page 29
- “Starting a Miniroot Installation” on page 30
- “Starting a Live Installation” on page 36

Determining the Installation Method

If the software product that you plan to install contains subsystems that require a miniroot installation, you must shut down the target system and install them from the miniroot; you cannot install these products by a live installation (see “Software Installation Methods” on page 10). To determine whether the software that you plan to install requires a miniroot installation, read the product release notes. Release notes list all subsystems in the product that require miniroot installation.

A miniroot installation is also when

- the target is being upgraded from a pre-6.3 version of IRIX to IRIX 6.3 or later
- the target has software problems that prevent successful live installations
- the target’s system disk is new or empty
- recovering from severe system crashes (system recovery)

Preparations Checklist

Complete the following preparations before installation:

1. Determine the appropriate installation method for your product.
See “Determining the Installation Method,” above.
2. Back up the target system.
Backing up essential files on the target system is not normally required but is strongly recommended as protection against unforeseen problems. See “Backing Up the Target Systems” on page 22 for details.
3. If the software distribution is located on a remote server, make these preparations:
 - Verify that the guest account or an alternate account is available to Inst on the remote distribution server. See “Configuring an Installation Account” on page 18 for details.
 - Jot down the host name and pathname of the distribution directory on the remote server. If you plan to do a miniroot installation, also jot down the network address of the remote server. See “Getting the Server’s Name and Address” on page 24 for instructions.
4. If you plan a miniroot installation, make these additional preparations:
 - Enable BOOTP forwarding on any routers that operate between the installation server and the target system. See “Enabling BOOTP Forwarding on Routers” on page 16 for instructions.
 - Enable TFTP access to the remote distribution server. See “Enabling TFTP Access on an Installation Server” on page 17 for instructions.
 - If your distribution source is a locally mounted CD, be prepared to specify the controller and unit number of the CD-ROM drive. See “Getting CD-ROM Device Numbers” on page 23 for instructions.
 - Be prepared to supply the hostname and IP address of the target. See “Getting the Target’s Name and Address” on page 24 for instructions.

About Miniroot Installations

For installations where fundamental IRIX services, such as filesystem management, are either unavailable or unreliable, a special installation method is required. This method, known as a *miniroot installation*, relies on services in the target's programmable read-only memory (PROM) to transfer special installation tools (see Figure 1-1 on page 6) from the distribution source to the target. This transfer is referred to as *loading the miniroot*.

Loading the miniroot is a two-phase process: first the *sash*, or standalone shell, is transferred from the source to the target system for execution; then, the *sash* fetches the miniroot from the source and copies it to the swap partition of the disk. The *sash* and miniroot are frequently loaded from a locally-mounted CD, but they can also be loaded from a network location using the Internet Bootstrap Protocol (*bootp*). A network host containing the *sash* and miniroot for various workstation and server architectures is referred to as a *boot server*.

Caution: The miniroot is copied from the distribution source to partition 1 of your system disk, which is normally the swap partition. If you have customized your system in such a way that partition 1 contains valid data, be sure to back up the data in partition 1 before the miniroot installation to avoid losing it.

By default, the miniroot restricts the target to installation operations and the target operates like an ASCII terminal, even on graphics systems. These restrictions preserve the integrity of the operating system while it is being updated.

Two methods are available for performing miniroot installations. The first method is used when only one target system is involved and the installer uses the target system directly. Instructions for using this method are given in "Starting a Miniroot Installation" on page 30. The second method installs any number of target systems from any network location.

Starting a Miniroot Installation

To start a miniroot installation, invoke the System Maintenance menu on the target system and load the miniroot from the distribution source. The procedure that you use to load the miniroot differs for local and remote distribution sources. Use one of the following procedures, depending on the location of your distribution source, to start a miniroot installation:

- “Loading From a Local CD,” below
- “Loading From a Remote Distribution Source” on page 34

Loading From a Local CD

The procedure below explains how to load the miniroot from a distribution CD that is mounted locally on the target. This procedure also explains the informational messages that you should check when `Inst` is invoked from the miniroot load.

Note: If you are using an ASCII console to perform this procedure, prompts and menus appear as text on your screen.

1. Shut down the target system.

You can shut down the target system by selecting System Shutdown from the System tool, or you can shut it down from an IRIX shell. Perform the shutdown as superuser:

```
% su -  
Password:  
# shutdown
```

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

```
Shutdown started. day time year  
Broadcast message from root (ttq1) on hostname day time year
```

This message is followed by a directive to all users to log off the system. Then this confirmation prompt is posted:

```
Do you want to continue with the shutdown (yes/no) [no]?
```

2. Confirm the shutdown.

Type **yes** to confirm the shutdown. After your entry, you see messages similar to Figure 3-1.

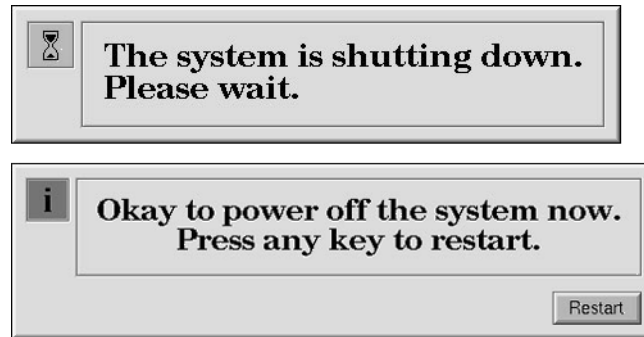


Figure 3-1 System Shutdown Messages

3. Begin the restart sequence.

Select *Restart* or press any key to begin the system startup process. After your entry, you will see a message similar to the one Figure 3-2.

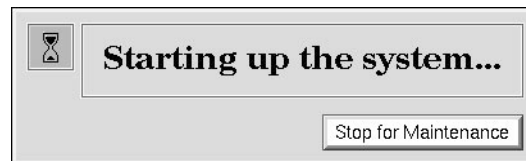


Figure 3-2 System Restart Message

4. Stop the system for maintenance.

When you see the message shown in Figure 3-2, select *Stop for Maintenance* or press Esc to go to the System Maintenance menu. The System Maintenance menu is shown in Figure 3-3.



Figure 3-3 The System Maintenance Menu

5. Choose *Install System Software* from the System Maintenance menu.

Click the *Install System Software* icon to display a selection of distribution sources. If you are using an ASCII console, type the number of the *Install System Software* selection to display a list of distribution choices.

After your selection, a menu of distribution selections is displayed.
6. Specify the location of the installable software.

Select the local CD-ROM icon to specify the location of the software distribution. If you are using an ASCII console, enter the number of the local CD-ROM selection.
7. Insert the first CD.

Be sure to insert the CD containing the installation tools into the caddy with the label side up. Then insert the caddy into the CD-ROM drive.
8. Load the miniroot and run *Inst*.

Click the *Continue* button or press *Enter* to start the miniroot load and run *Inst*. These messages appear after your entry:

```
Obtaining installation tools
Copying installation tools to disk
```

9. Wait for the miniroot to be copied.

Copying the miniroot to the swap partition of the target takes several minutes. During that time, a status bar appears to report the progress of the copy (rows of dots appear on ASCII console screens). When the copy is completed, some informational messages are posted and the Inst Main menu is displayed. The informational messages are similar to these:

```
IRIX Release version cpu Version longversion System V
Copyright 1987-1998 Silicon Graphics, Inc.
All Rights Reserved.
```

```
ipg0: missing
enp0: missing
fxp0: missing
```

These messages are normal and do not indicate a problem; one line is posted for each type of device that is supported by the kernel and not detected on the target. After the device messages, startup messages are displayed:

```
Creating miniroot devices, please wait...
```

```
Current system date is date
```

```
Mounting file systems:
```

```
    /dev/miniroot           on /
    /dev/dsk/dks0d1s0       on /root
    /root/dev/usr           on /root/usr
```

```
Invoking software installation.
```

10. Reset the system date, if needed.

Notice that the startup messages include a posting of the system date. If the date is off by more than a few minutes, reset it with this command (see “Changing the Network Time-out Value” on page 125 if you need more information):

```
Inst> admin date mmdhmmyy
```

To recheck the date, display it again with this command:

```
Inst> admin date
```

11. Check the distribution source that is specified for the installation.

The local CD distribution directory, */CDROM/dist*, should be posted as the distribution specification in a message above the Inst Main menu (see Figure 4-2 for an illustration of the Main menu):

```
Default distribution to install from: source
For help on inst commands, type "help overview".
```

The miniroot is successfully loaded on the target system and `Inst` is invoked. Go to Chapter 4, “Using the Main Menu,” to install the software.

Loading From a Remote Distribution Source

The procedure below explains how to load the miniroot from a CD or directory located on a remote server (if you plan to install from a local CD, see “Loading From a Local CD” on page 30). This procedure also explains the informational messages that you should check when `Inst` is invoked from the miniroot load.

Note: If you are using an ASCII console to perform this procedure, prompts and menus appear as text on your screen.

1. Follow steps 1 through 5 of “Loading From a Local CD” on page 30.
2. Specify the location of the installable software.

Select the *Remote Directory* icon to specify the location of the remote distribution. If you are using an ASCII console, enter the number of the remote directory selection.

3. Specify the hostname of the installation server.

After you select *Remote Directory*, you are prompted to enter the name of the remote host that contains the distribution. Specify the hostname of the installation server, as shown in this example:

```
Enter the name of the remote host: installation_server
```

4. Specify the pathname of the distribution source.

After you enter the name of the installation server, you are prompted to enter the name of the remote distribution directory. If your distribution source is a CD on the remote server, your entry resembles the example shown below. In your entry, substitute the mount directory of the CD-ROM drive for *CDdir*:

```
Enter the remote directory: CDdir/dist
```

If your distribution source is a directory on the remote server, your entry will resemble the example shown below. Substitute the pathname of the distribution directory for *distdir*:

```
Enter the remote directory: distdir
```

After your entry, you will see a message like the examples below:

```
Remote directory CDdir/dist from server installation_server.  
Remote directory distdir from server installation_server.
```


5. Click the *Install* button or press Enter to start the miniroot load.

After your entry, the installation tools and miniroot are copied to the target. These messages appear while the copy is in progress:

```
Obtaining installation tools from server:distdir
Copying installation tools to disk
```

6. Wait for the miniroot to be copied.

Copying the miniroot to the swap partition of the target takes several minutes. During that time, a status bar appears to report the progress of the copy (rows of dots appear on ASCII console screens). When the copy is completed, some informational messages are posted and the Inst Main menu is displayed. The informational messages are similar to these:

```
IRIX Release version cpu Version longversion System V
Copyright 1987-1994 Silicon Graphics, Inc.
All Rights Reserved.

ipg0: missing
enp0: missing
fxp0: missing
```

These messages are normal and do not indicate a problem; one line is posted for each type of device that is supported by the kernel and not detected on the target.

After the device messages, startup messages are displayed:

```
Creating miniroot devices, please wait...
Current system date is date
Mounting file systems:
    /dev/miniroot          on /
    /dev/dsk/dks0d1s0     on /root
    /root/dev/usr         on /root/usr

Invoking software installation.
```

7. Reset the system date, if needed.

Notice that the startup messages include a posting of the system date. If the date is off by more than a few minutes, reset it with this command (see “Changing the Network Time-out Value” on page 125 if you need more information):

```
Inst> admin date mmdhhmmyy
```

You can display the date again with this command:

```
Inst> admin date
```

8. Check the distribution source that will be used for the installation.

The location of the distribution source that you specified in the miniroot load procedure is posted in a message above the Inst Main menu (see Figure 4-2 on page 40 for an illustration of the Main menu):

```
Default distribution to install from: source  
For help on inst commands, type "help overview".
```

The miniroot is successfully loaded on the target system and Inst is started. Go to Chapter 4, “Using the Main Menu,” to install software.

Starting a Live Installation

It is not necessary to shut down the target system to perform a live installation. However, it is always advisable to warn other users to shut down applications that will be affected by the installation. During a live installation, Inst creates copies of some files that are held open by the operating system or by applications. For this reason, live installations usually require extra disk space temporarily.

To install software, you must start Inst as the superuser on the target system. If you start Inst as a regular user, you are assigned read-only permissions, preventing you from making any changes to the installed software.

Note: If you are running on your desktop, you can also use SoftwareManager. See the *Personal System Administration Guide*, and the *swmgr(1M)* reference page for more information.

Use this command to start Inst from the command line of an IRIX shell on the target:

```
% su  
Password:  
# inst
```

After your entry, the default location of the distribution is posted and the Inst Main menu is displayed (see Figure 4-2 on page 40).

Note: When starting a live installation, you can use arguments to the *inst* command to modify the way that Inst operates. Some of these arguments are described in Chapter 6, “Inst for Experts.” A complete list of *inst* arguments is given in the *inst(1M)* reference page.

Using the Main Menu

This chapter describes how to install software using the Inst Main menu. It explains how to use Main menu commands in a typical installation session and what to do after the installation is done to put the target back into service. Use the instructions in this chapter to complete the installation after you have performed the procedures in Chapter 3, “Starting an Installation.”

This chapter contains these sections:

- “The Software Installation Procedure” on page 38
- “If Inst Is Interrupted” on page 57
- “Managing Configuration Files” on page 60

If you prefer to use a graphical user interface and Software Manager is installed on the target system, you can access Software Manager from the System toolchest and use it instead of Inst. See the software installation instructions in the *Personal System Administration Guide* for how to use Software Manager.

The Software Installation Procedure

The procedure in this section shows you how to perform a software installation using a particular sequence of commands from the Inst Main menu. The order in which the commands are described illustrates one of many command sequences that might be used to do an installation. As you become familiar with Inst, you can vary the command order, repeat commands, use additional keyword arguments, and enter commands from other menus during installation sessions (Chapter 5, “Using Supplementary Menus,” and Chapter 6, “Inst for Experts,” contain information about additional Inst features).

Note: If you are an experienced Inst user, you might prefer to use the “Condensed Installation Instructions” on page 177 instead of this procedure.

This procedure partitions the installation into seven steps. Each step is explained in the subsections that follow.

1. Start Inst.
2. Specify the distribution source(s) using the *from*, *open*, and *close* commands.
3. Display software information using the *list* commands.
4. Make your software selections with *install*, *remove*, and *keep* commands.
5. Launch the installation using the *go* command.
6. Resolve any conflicts (*conflicts* command) or errors.
7. Quit the installation session.

Figure 4-1 illustrates the steps in the installation process. In this figure, the automatic initialization function is shown at step 2. However, initialization can occur at other steps in the session (see “Automatic Initialization Functions” on page 42 for details).

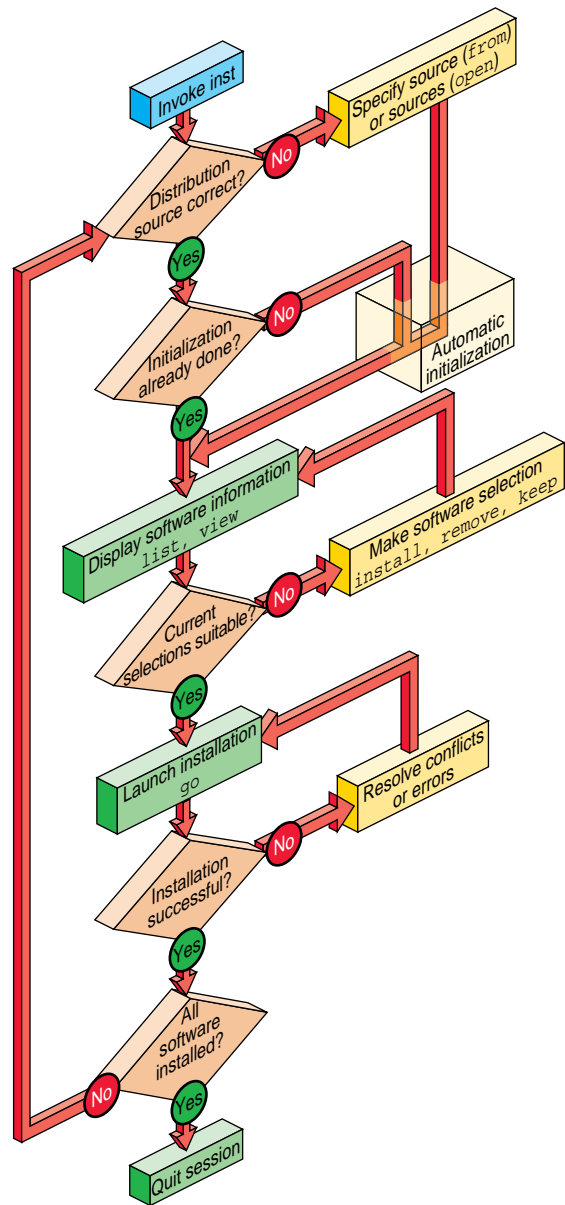


Figure 4-1 Summary of the Installation Procedure

Step 1: Invoking Inst

For miniroot installations, which are required whenever you install software that is fundamental to the IRIX operating system, Inst is automatically started when the miniroot is loaded (see “Starting a Miniroot Installation” on page 30). For live installations, Inst is started from the command line of an IRIX shell (see “Starting a Live Installation” on page 36). After Inst is started, it displays the default distribution source and the Main menu, as shown in Figure 4-2.

```
Default distribution to install from: magnolia:/dist/6.5/c
```

```
For help on inst commands, type "help overview".
```

```
Inst 3.4 Main Menu
```

1. from [source ...]	Specify location of software to be installed
2. open [source ...]	Specify additional software locations
3. close [source ...]	Close a software distribution location
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]	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 Menu
13. admin ...	Go to the Administrative Commands Menu
14. quit	Terminate software installation

Figure 4-2 The Inst Main Menu

Step 2: Specifying the Source

Specifying a distribution source is not always necessary during an installation procedure, for frequently the default distribution is appropriate. By default, Inst uses the distribution that was previously specified as the source of installable software for a live installation. For miniroot installations, Inst expects to use the distribution from which you loaded the miniroot. The default source is posted in a message above the Main menu when Inst is started (see the final step in “Loading From a Local CD” or “Loading From a Remote Distribution Source” in Chapter 3).

If the default source is not appropriate, however, use the *from* command to replace the current distribution source; use the *open* command to specify additional sources of installable software. The *close* command closes any distribution that you specified with either the *from* or the *open* command. You can use the *from*, *open*, and *close* commands as many times as necessary during an installation session.

Note: When Inst posts multiple distributions in *from*, *open*, and *close* listings, it numbers each item in the list. You can use this number as a command argument instead of typing the entire distribution pathname. For example, *close 2 4 7* closes the three distribution sources numbered 2, 4, and 7 in the distribution list.

Using the *from* Command

Use the *from* command to replace the current distribution source with one or more different sources. Your entry can take one of the forms of *from* listed below (also see “Alternative Ways to Specify a Distribution” on page 83):

from Enter the *from* command with no arguments to display a selection of former distributions from which to choose; type the number of a selection to specify it as the distribution. If the distribution that you want does not appear in the list, enter an alternate form of the *from* command (below) to specify the distribution.

from *pathname* ...

from *server:pathname* ...

Add a pathname argument to *from* to specify a distribution that is local to the target system (usually a CD-ROM drive) or NFS mounted on the target system. If the distribution is on a remote system, the pathname is preceded by the remote hostname and a colon. You can specify more than one distribution source in your entry; use a space to separate each distribution.

These examples illustrate *from* command entries for a local distribution, a remote distribution, and multiple distributions:

```
from /CDROM/dist
from easyboy:/d1/IRIX_6.5/dist
from easyboy:/d1/IRIX_6.5/dist sparkplug:/dist/c
```

from none Specify *none* to display the list of software installed on the target; no distribution is selected in this case. This form of the *from* command is useful for browsing and removing software from the target system.

Using the open Command

Use the *open* command to add a distribution source to the sources that are currently available to Inst. The *open* command requires the location of the additional distribution as an argument. For example, if *easyboy:/d1/IRIX_6.3/dist* is the current distribution source, the command in example Example 4-1 opens the distribution at *easyboy:/CDROM/dist* and makes it available for installation.

Example 4-1 The open Command

```
Inst> open easyboy:/CDROM/dist
```

When you use the *list* command (see “Step 3: Displaying Software Information,” which follows), the installable components from the newly opened source are added to the listing following the components from the original source. In the Example 4-1 case, the listing from *easyboy:/CDROM/dist* would follow the listing from *easyboy:/d1/IRIX_6.3/dist*.

Also see “When Multiple Distributions Are Open” on page 47 for more information.

Automatic Initialization Functions

Notice in Figure 4-1 that Inst performs an automatic initialization sequence during the installation procedure. Initialization is triggered by the first command of a session that requires Inst to collect information about the distribution or target software. In Figure 4-1, the trigger is the *from* command. During initialization, Inst obtains historical data about previous installations and reads the product descriptions on the distribution. It also checks for software dependencies among subsystems, makes preliminary installation selections, calculates their sizes, and checks disk availability on the target.

As this initialization occurs, you see this message:

```
Reading product descriptions .. 100% Done.
```


When initialization is complete, the `Inst>` prompt is posted and you can continue with the installation session.

Note: If you do not enter the *from* command in this step, Inst performs initialization before executing the next command that requires data collection.

Step 3: Displaying Software Information

Displaying software information is an optional step in the installation procedure. You can omit this step if you are already familiar with the products that you want to install or if you plan to accept the selections made during initialization (see “Accepting Automatic Selections” on page 50). For most operating system installations, however, this step is necessary to evaluate distribution and target software and do progress checks during the selection process.

By default, the *list* command displays a list of all subsystems in the distribution inventory, but you can also display a complete list of software on the target using the *view* command (see “Using the View Commands Menu” on page 64 for additional ways of using *view* to control *list* displays). As you select software for installation or removal (step 4 of this procedure), *list* displays are automatically updated to reflect your selections.

Using the list Command

Use the *list* command with no arguments to display the distribution list (if you did not enter the *from* command in step 2, Inst performs initialization before executing the *list* command). You can use the names of products and their components, metacharacters, and other keywords as arguments to *list* commands to save time (see “Using Command Shortcuts” on page 85 for details).

Example 4-2 illustrates a *list* entry and the legend that precedes *list* displays.

Example 4-2 The list Legend

```
Inst> list
View:      distribution
Status:    N=New, U=Upgrade, S=Same, D=downgrade
Selection: i=install, r=remove, k=keep

Subsystem Type(s) [bdrpc]: b=reBoot needed, d=Default, r=Required, p=Patch, c=client only
```

As Example 4-2 shows, the *list* legend provides information about software components and about actions taken during the installation session:

- *View* identifies the location of software in Inst displays: target or distribution.
- *Status* mnemonics convey comparison information between software on the target and software on the distribution software.
- *Selection* mnemonics convey information about installation actions that have been specified either during automatic initialization or by the user.
- *Subsystem Types* mnemonics are the only mnemonics to appear to the right of the software name in a *list* display; they provide information about distribution software requirements and designations.

Note: If the distribution contains patches, patch mnemonics are also shown. See “Interpreting Patch Information in list Displays” on page 100 for details.

Example 4-3 illustrates sample *list* output of a distribution inventory.

Example 4-3 Distribution Software Posted by the list Command

```
S bvo.sw.bvo                0 VGX Broadcast Video Option Software
S bvo.sw.diag              0 VGX Broadcast Video Option Diags
i U compiler_eoe.sw.cpp [bdr] 0 Source Code Preprocessor
i U compiler_eoe.sw.lboot    43+ Kernel lboot Software
D desktop_tools.sw.tools [d] 0 Desktop Tools
N onc3_eoe.sw.cachefs       0 ONC3 6.3 Cache File System
```

Frequently, *list* displays contain several screens of information. To display more information, press Enter at the *more* prompt to see a new line and press the space bar to see a new screen. You can also quit the display before reaching the end by typing *q* at the *more* prompt.

Displaying Software Installed on the Target

Use these commands to change the view and display target software:

```
Inst> view targ
Inst> list
```

Example 4-4 shows sample *list* output of target software (the legend is omitted from this display).

Example 4-4 Target Software Posted by the *list* Command

```

I bvo.sw.bvo                0 VGX Broadcast Video Option Software
I bvo.sw.diag              0 VGX Broadcast Vidio Option Diags

u I compiler_eoe.sw.cpp [bdr] 0 Source Code Preprocessor
u I compiler_eoe.sw.lboot    43- Kernel lboot Software

I desktop_tools.sw.tools [d] 0 Desktop Tools

R showcase.sw.data [d]      0 Necessary Data Files
R showcase.sw.showcase [d]  0 IRIS Showcase 3.3 Software

```

Use this command to return the view to the distribution:

```
Inst> view dist
```

Interpreting *list* Output

A *list* display contains several columns of information about each software component in the display. Some columns convey information by means of mnemonic tags, which are defined in the *list* legend (see Example 4-2). Columns appear empty when a tag is unassigned or when the tag is masked by the current view setting (see “The target and distribution Commands” on page 65 for more information).

The columns in *list* displays contain this information:

- Column 1 A pending selection on the software component, if a selection was made. This column may contain any item listed in the *Selection* line of the *list* legend.
- Column 2 The status of the software component relative to its corresponding component on the target or distribution. This column contains an item listed in the *Status* line of the *list* legend (see Example 4-2).
- Column 3 The name of the software component. When software subsystems are displayed, the name may be followed by one or more special designations from the *Subsystem Type* line of the *list* legend (see Example 4-2).

- Column 4 For software items being installed (i) or removed (r), a comparison between the disk space requirements of the distribution and target versions of the software. The amount, which is displayed in kilobytes by default, is followed by a plus sign if additional space is required or a minus sign if there is space free. If items are not selected for installation or removal or if there is no net change, the size shown is zero.
- Column 5 A brief description of the software component. The content of this description varies slightly with each product.

These items from Example 4-3 further illustrate the information provided about distribution software by the *list* command:

- S bvo.sw.bvo The distribution version of the *bvo.sw.bvo* subsystem is the same (S) as the version on the target; no action is to be taken on this subsystem (the first column is blank).
- i U compiler_eoe.sw.cpp [bdr] The distribution version of *compiler_eoe.sw.cpp* is an upgrade (U) to the version that is currently on the target. This subsystem is selected for installation (i). Installing *compiler_eoe.sw.cpp* requires a reboot (b) of the target, the manufacturer has selected this subsystem as a default (d) for installation; *compiler_eoe.sw.cpp* is required (r) for system operation.
- i U compiler_eoe.sw.unix...26- The distribution version of *compiler_eoe.sw.unix* is an upgrade (U) to the version that is currently on the target. This subsystem is selected for installation (i); the distribution version is 26 kilobytes smaller than the corresponding version on the target.
- D desktop_tools.sw.tools [d] The *desktop_tools.sw.tools* subsystem in the distribution is a downgrade (D) of the corresponding subsystem on the target. Because the target version of this subsystem is newer, no action is to be taken on this subsystem, even though the manufacturer recommends it as a default (d) installation selection.

The information in Example 4-4 indicates that most products from the distribution are currently installed (I) on the target but that the *showcase* subsystems have been removed (R). One distribution product from Example 4-3, *onc3_eoe*, does not appear in the target list in Example 4-4. Also notice that items marked for installation (i) in the distribution list (Example 4-3) are marked for upgrading (u) in the target list (Example 4-4).

When Multiple Distributions Are Open

When multiple distributions are open, `list` posts the name of the distribution at the top of the distribution contents. For example, this listing indicates that two distributions are open, one on server `pearl` and one on server `cameo`:

Example 4-5 List Output for Multiple Open Distributions

```
pearl:/6.5/options/SoftWindows95:
i U SoftWindows95.man.eoe [d]      0  SoftWindows 95 Man Pages
i U SoftWindows95.man.relnotes [d] 0  Release Notes
i U SoftWindows95.sw.eoe [d]     14+ SoftWindows95 core
i N SoftWindows95.sw.help [d]    5+  SoftWindows95 Help files

cameo:/apps/acrobat_3.0:
i U acrobat.man.man [d]           0  Acrobat reader Manual Pages, 3.0
i U acrobat.man.relnotes [d]     1-  Acrobat reader Release Notes, 3.0
i U acrobat.sw.reader [d]       2245+ Acrobat reader, 3.0
```

Interpreting the Disk Space Summary

At the end of each listing, the `list` command posts a disk space summary similar to those shown in Example 4-6. The summary identifies the target filesystems to receive the selected software, estimates the additional space (or space savings) that will result if the selected software is installed, calculates the space overhead required by installation processing, and reports the amount of space that will be available on the target filesystems after the installation. Estimates in disk space summaries are based on the current installation selections.

Example 4-6 Disk Space Summary

```
Disk Space summary (Kbytes)      /
Current free space                564432
- Selections net change           1276+
- Temporary inst overhead        53+
= Minimum free during install    563103
Final projected free space       563156
```

Step 4: Making Software Selections

Three types of actions can occur on a target during an installation: Distribution software is installed, target software is removed, and some target software is kept in its present condition. In this step, you specify which of these actions should be taken against selected software when the installation is performed (step 5 of this procedure). The *install*, *remove*, and *keep* commands select software for the designated action.

In most cases, the selection step is needed to adjust software for the disk capacity of the target and the needs of its users. However, in some cases it is possible to accept the automatic selections that Inst makes during initialization. These selections are described in “Accepting Automatic Selections” on page 50.

Note: If the automatic selections described on page 50 are suitable for your circumstances, you can accept these selections and go directly to step 5 of this procedure.

Using the *install*, *remove*, and *keep* Commands

The *install*, *remove*, and *keep* commands select software products and their components for a particular action. No action is taken until you enter the *go* command, so you can change your selections as many times as necessary before entering *go* (described in “Step 5: Launching the Installation” on page 51).

Note: You can use the *step* command to display items on the distribution individually and make your selections as each item is displayed. See “Using step to Make Selections” on page 90 for details.

When you make your selections, keep these points in mind:

- Usually, Inst has already made some installation selections automatically (see “Accepting Automatic Selections” on page 50). Use the *install* command to select software that has not already been selected.
- In some cases, you might not want a selected software component installed or you might want to retain the currently installed version rather than upgrade it. When this is the case, use the *keep* command to preserve the current condition on the target.
- During the installation, Inst automatically removes target software that will be upgraded with software from the distribution (illustrated by Example 4-4). It is not necessary to use the *remove* command on these products—use the *remove* command only on currently installed software that you no longer want on the target.

- Use the *list* command periodically to check the selection status of products and product components during the selection step (see “Step 3: Displaying Software Information” on page 43).
- It is sometimes useful to alternate the selection process with the launch process (see “Step 5: Launching the Installation” on page 51), selecting and installing a subset of software in each cycle. In this case, alternate the selection commands with the *go* command.

Simple Selection Command Entries

In their simplest form, the *install*, *remove*, and *keep* commands contain one argument that identifies the product, image, or subsystem on which the action will be taken. If you specify a product or image as the argument to the *install*, *remove*, or *keep* commands, the action is taken on all subsystems of the product or image.

These sample entries explain the effects of selection command entries:

```
Inst> install sysmon
```

Install all parts of the product *sysmon*, which is either a new product for the target or an upgrade to the installed version.

```
Inst> install prereqs
```

Automatically resolve conflicts by installing any prerequisites for selected software.

```
Inst> remove InPerson.books
```

Remove the currently installed *InPerson.books* image from the target.

```
Inst> keep DeltaCC.sw.backend
```

Retain the current version of *DeltaCC.sw.backend* that is installed on the target; if *DeltaCC.sw.backend* is not installed, prevent installation of the distribution version.

```
Inst> keep conflicting
```

De-select all subsystems causing conflicts.

You can use multiple arguments in *install*, *remove*, or *keep* commands, as shown in Example 4-7.

Example 4-7 Multiple Arguments to the install, remove, and keep Commands

```
Inst> install sysmon cadmin mmail desktop_eoe dmedia_eoe quickt  
ime_dev imgtools.sw showcase  
Inst> keep InPerson.books InPerson.sw  
Inst> remove DeltaCC.sw.backend DeltaCC.man.relnotes
```

Using Arguments in Selection Commands

You can use the name of a distribution, products and their components, metacharacters, and keywords as arguments to the *install*, *remove*, and *keep* commands to save time (see “Using Command Shortcuts” on page 85 for more information). These sample entries illustrate the use of shortcuts in selection commands (see “Using Command Shortcuts” on page 85 for more information):

```
Inst> install required
```

Install all subsystems that are required for optimum system operation.

```
Inst> remove distribution_name: *.man
```

Remove all images with a .man extension from the specified distribution. The distribution name must end with a colon (:).

```
Inst> keep I eoe*
```

Keep all *eoel* subsystems that are currently installed on the target (do not install upgrade versions from the distribution).

Checking Your Selections With list

You can enter a *list* command during the selection process whenever the `Inst>` prompt is displayed. The first column of the listing and the disk space summary information reflect your selections and their effects on disk space.

Accepting Automatic Selections

Recall from step 3 that Inst automatically makes preliminary selections during initialization (see also “Automatic Initialization Functions” on page 42). If you accept the automatic selections, you can omit step 4 and go on to “Step 5: Launching the Installation” on page 51.

Use this command to display the list of software that is selected for installation:

```
Inst> list i
```

Example 4-8 illustrates a portion of the display that was generated by the previous *list* command. Notice that the display concludes with an estimate of space requirements, which may help you decide whether to accept the automatic selections.

Example 4-8 List of Subsystems Selected for Installation

```
i U eoe.sw.gfx_lib bdr      19+ Graphics Execution Libraries
i U eoe.sw.irix_lib bdr    991+ IRIX Execution Libraries
i U eoe.sw.unix bdr       6674+ IRIX Execution Environment
i U eoe.sw.cdrom d        241+ CD-ROM Support
i U eoe.sw.crypt d        25+ Security Administration Utilities
i U eoe.sw.gltools d       80+ Graphics Library Tools

Disk Space summary (Kbytes):
Current free space          564432
- Selections net change     64793+
- Temporary inst overhead   53+
= Minimum free during install 499586
Final projected free space  499639
```

Step 5: Launching the Installation

The selections that you made in step 4 are processed when you launch the installation with the *go* command. You can process any number of selections; it is not necessary to complete all your selections before you enter *go*. The order in which you made selections (step 4) has no effect on the order in which they are processed.

Note: Inst reports a conflict during *go* processing if you failed to include any prerequisite software in a selected subset (see “Step 6: Resolving Conflicts” on page 52).

As the *go* command executes, Inst determines whether your selections contain incompatibilities, missing prerequisites, space shortages, or other errors that might make the new software configuration unsuitable for the target. To keep you informed of events, Inst posts status messages during *go* processing (see Example 4-9).

Example 4-9 Successful Installation Messages

```
Inst> go
Reading fileset information..
Pre-installation check .. 100% Done.
Checking space requirements ...
Installing/removing files .. 44%
Upgrading selected pv.sw subsystems
Installing/removing files.. 100% Done.
Running exit commands .. 100% Done.
Checking dependencies .. 100% Done.
Installations and removals were successful.
You may continue with installations or quit now.

Inst>
```

Notice that Example 4-9 contains a success message:

```
Installations and removals were successful
```

When you see this message, you can either continue the installation session or go directly to “Step 7: Quitting the Installation Session” on page 55. If you receive an error message instead of a success message, complete “Step 6: Resolving Conflicts” before going on to step 7.

Step 6: Resolving Conflicts

The installations and removals that you specified in step 4 are not performed if conflicts or error conditions are detected during *go* processing. Conflicts occur if you select software that depends on prerequisites that you did not select or if a selection is incompatible with other selections or with installed software. Errors other than conflicts that occur during *go* command processing are resolved with routine corrective actions (see Appendix B, “Troubleshooting Installations,” for additional error-handling information).

Using the `conflicts` Command

If Inst detects a conflict in the software that you selected, it posts a `conflicts` message after you enter the `go` command. Conflict error messages are followed by a description of the conflicts and recommended actions that you can safely perform to resolve the conflicts. You use the `conflicts` command to select a suggested resolution. You can also enter the `conflicts` command periodically during the selection process to display any conflicts that might be accumulating as a result of your selections.

Example 4-10 illustrates a message describing two conflicts.

Example 4-10 Conflicts Message

```
ERROR: Conflicts must be resolved.
Movie Playback, Recording, Conversion cannot be installed because of
missing prerequisites:
  1a. Do not install Movie Playback, Recording, Conversion
  1b. Also install Compression Execution Environment
DLPI Execution Environment cannot be removed because other products
depend on it:
  2a. Do not remove DLPI Execution Environment
  2b. Also remove 4DDN Software
```

To resolve a conflict, enter the `conflicts` command and your choice of resolutions as command arguments, as shown in Example 4-11.

Example 4-11 Entry to Resolve Conflicts

```
Inst> conflicts 1b 2a
```

When you have successfully resolved all conflicts, Inst posts a success message. Enter the `go` command after the message to process any new selections.

```
No conflicts
Inst> go
```

If the `go` command returns a success message (see Example 4-9), you can continue with the session or go on to “Step 7: Quitting the Installation Session” on page 55.

Resolving Other Errors

If *Inst* is unable to execute *go* for reasons other than conflicts, it posts error messages that suggest the cause of the error and presents choices for your next action. Errors can occur during preinstallation checking, while software is being installed and removed, or during exit operations. If you need additional information to correct an error, see Appendix B, “Troubleshooting Installations,” which describes the errors that occur in each phase of the installation process and suggests how to handle them.

Errors Caused by Read-Only Directories

A common error in preinstallation checking occurs if software is to be installed into or removed from a directory with read-only permission, which makes the directory inaccessible to the *inst* utility. Example 4-12 illustrates the message that is displayed during preinstallation checking when this error occurs.

Example 4-12 Error With Directory Permissions

```
Inst> go
```

```
ERROR: Cannot begin processing the following subsystems because they must
install/remove files in read-only directories that are on your system or on a
remote NFS server. Okay to continue without these subsystems?
```

If you continue, *Inst* deselects the subsystems that require access to the read-only directories listed in the message, then it continues with *go* processing. However, if deselecting these subsystems will generate conflicts with other subsystems that are being installed or removed, *Inst* posts one of the directory-permissions errors shown on page 213 of Appendix B, “Troubleshooting Installations.”

Errors that Display the Interrupt Menu

During *go* processing, you see status messages like those shown in Example 4-13. If an error occurs that requires you to take further action, *Inst* displays the Interrupt Menu, also shown in Example 4-13.

Example 4-13 Error in Installation Processing

```
Inst> go
Pre-installation check
Installing/removing software
Upgrading selected prod1.man subsystems
Installing new versions of selected prod1.sw subsystems
ERROR : An error occurred while Installing new versions of selected
prod1.sw subsystems
Archive /swmgr/products/prod1/images/prod1.sw is corrupt

Error/Interrupt Menu
 1. retry           Retry the failed operation
 2. stop           Terminate current command
 3. continue       Continue current command
 4. set[preferences] List all preferences or set/clear a preference
 5. help [topic]   Get help in general or on a specific word
 6. sh [cmd]       Escape to a shell or run a command
 7. shroot [cmd]   Escape to a chrooted shell or run a command

Interrupt>
```

The version of the Interrupt menu in Example 4-13 includes a “retry” choice, which causes Inst to attempt the failed operation again (commonly, these failures are due to a network timeout). Whenever the Interrupt Menu is displayed, check the ERROR line above the menu for a suggested reason for the failure, take the appropriate corrective action (if any is required), then retry the operation. See “Using the Interrupt Menu” on page 79 for information on other Interrupt Menu selections (notice that the Interrupt Menu in Figure 5-3 does not contain the “retry” choice).

Step 7: Quitting the Installation Session

You can leave an installation session at any time by issuing the *quit* command at the Inst prompt.

```
Inst> quit
Requickstarting ELF files (see rqsall(1))...100% Done.
Automatically reconfiguring the operating system.
```

If Selections Are Pending

If any installation or removal selections are pending when you enter the *quit* command, Inst notifies you:

```
There are products marked for installation or removal.  
Do you really want to quit?(y/n)
```

If you enter *y*, the session ends and your pending selections are not processed. Enter *n* if you want to continue the session.

If Conflicts Are Unresolved

In some cases, there may be unresolved conflicts when you enter the *quit* command. For example, unresolved conflicts occur when you try to end a session without installing all software that is required for an operating system upgrade.

If there are unresolved conflicts when you enter the *quit* command, Inst posts the message shown in Example 4-14.

Example 4-14 Conflicts Message When Quitting Inst

```
ERROR: INCOMPATIBLE SUBSYSTEMS INSTALLED
```

```
Some software that was already installed on your system prior to this  
session is incompatible with the new software. The conflict messages above  
(or below) list the incompatible software. You can either:
```

- Insert another CD or specify another distribution directory that contains compatible versions of software (for example, if you are in the middle of an operating system upgrade, insert the next CD that you received with the upgrade), then start the installation.
- Remove the incompatible software by making conflict resolution choices as shown above (or below).

This message is followed by a list of conflicts that must be resolved before you can end the session successfully. See “Step 6: Resolving Conflicts” on page 52 for information on conflict resolution.

Note: If you set the *rulesoverride* preference on (not recommended), the conflicts message varies from the one shown in Example 4-14. With *rulesoverride* on, the message identifies the unresolved conflicts and warns you that the installation will be performed despite remaining conflicts.

If Rebooting Is Needed

If you performed a live installation and installed products that require a system reboot, you see the message in Example 4-15.

Example 4-15 Quit Prompt With Reboot Message

```
Installations and removals were successful
You must reboot your system to complete the installation.
Inst>
```

If Inst Is Interrupted

If an installation session is interrupted by an abnormal event (such as a power failure), Inst saves a record of pending requests for product installations and removals in a *checkpoint restart file*. At the start of the next session, Inst posts a checkpoint restart notice that identifies the software distribution and lists the products on which actions are pending.

Completing a Checkpoint Restart

This checkpoint restart menu (see Figure 4-3) is displayed if an installation session is interrupted abnormally:

```
$ inst

A previous installation session was not completed successfully
-----
Software distribution location:server:pathname
Install sc.sw.sc

-----
1. Retry the previous installation session.
2. Restore the previous installation session and let me look at it.
3. Ignore the previous installation session.

Please enter a choice [1]:.
```

Figure 4-3 The Checkpoint Restart Menu

To proceed, choose one of the restart selections from this prompt. Checkpoint restart selections have these effects:

- If you select `Retry` the previous installation (the default), `Inst` attempts to restore the previous session and continue the installation.
- If you select `Restore` the previous installation session and let me look at it, `Inst` displays the `Inst>` prompt. Use the `list i` command to see what remains to be installed, and then enter the `go` command.
- If you select `Ignore` the previous installation session, the previous session is aborted (pending selections are disregarded) and `Inst` is started in a new session.

If a Checkpoint Restart Fails

Failures during a checkpoint restarts are frequently caused by four conditions, which are usually easily corrected:

- A missing software distribution

```
ERROR : The distribution server:pathname does not exist
```

This failure occurs if the distribution source that was saved in the checkpoint restart file no longer exists or is unavailable. For example, the distribution source may be inaccessible because of a network failure or because the distribution was moved.

If the distribution is missing and you select item 1, the error message is posted as soon as you make your selection. If the distribution is missing and you select item 2, the error message is posted when you try to list the distribution.

- Different software distribution

```
ERROR : Invalid product sc.sw.sc in selections file
```

This failure occurs if the location of the distribution source in the checkpoint restart file is the same but the products in the distribution are different. For example, you see this error if a CD other than the correct distribution CD is loaded in the CD-ROM drive or if the contents of the distribution directory were changed since the previous session.

If the software distribution is different from the previous installation and you select item 1, the error message is posted as soon as you make your selection. If the software distribution is different from the previous installation and you select item 2, the message is posted when you try to list the distribution.

- **Bad selections**

ERROR : Could not retry previous installation because of errors in the selections. Conflicts must be resolved.

This failure occurs when the installed software on the target or the product dependencies on the distribution has changed since the previous installation. The changes result in conflicts that prevent installation. If there are bad selections and you select item 1 to restart the installation, the installation is stopped and you see the error message.

After the message, you see the standard conflicts screen. If there are bad selections and you choose item 2 to restart the installation, you see only the conflicts screen. See "Using the conflicts Command" on page 53 if you need instructions on how to proceed.

- **Not enough disk space**

This failure occurs when the software that was selected for the previous installation requires more disk space than is available on the target. If there is not enough disk space and you select item 1 to restart the installation, you see this message:

ERROR : Could not retry previous installation because of errors in the selections. Installation requires too much space.

If you select item 2 to restart the installation, you see this message:

ERROR: Installation requires too much space.

To complete the installation, remove files from the target system or change your installation selections; then enter the *go* command.

Managing Configuration Files

Sometimes a distribution contains configuration files, which are either required or recommended for operation, that exist on the target in a modified form (an */etc/rpc* file, for example). When the target system contains modified configuration files, Inst preserves the modified files during the installation in one of two ways:

- If the new configuration file is required for operation, Inst copies the existing target version to a file of the same name, appending a *.O* (obsolete) extension to it (*/etc/rpc.O*, for example). Then it installs the new, required version of the file by its standard name (*/etc/rpc*).
- If the new configuration file is recommended (but not required), Inst leaves the existing file on the target untouched; it writes the recommended version to a file of the same name, appending a *.N* (new) extension to it. You can use *.N* files at your discretion.

For example, assume that the target contains a modified */etc/fstab* file before an installation and the distribution contains a new, recommended */etc/fstab*. After the installation, the original target version is still in */etc/fstab*; the new version is contained in */etc/fstab.N*.

Configuration File Anomalies

In most cases, Inst handles modified configuration files with predictable results: after installation, the target contains an installed version and an alternate version of the file. Normally, relevant portions of the alternate versions are subsequently merged into the installed version and the alternate version is removed. However, if you repeat an installation of the same software, the results may be different.

For example, it is possible to create both a *.O* and a *.N* version of the same configuration file. This happens when Inst creates a *.O* version of the file and you modify the newly installed version instead of merging and removing the *.O* version. During the repeat installation, Inst compares the installed file with the version on the distribution; when it finds that the two are different, it creates a copy of the distribution version, assigning it a *.N* extension.

As another example, assume that the *.N* version of a configuration file is merged with the installed version and erased. During the repeat installation, Inst detects differences between the installed file and the distribution version, so it creates a copy of the distribution version, assigning it a *.N* extension.

The Configuration File Notice

When a new version of a configuration file is created during an installation, Inst posts a message about the changed files after *go* processing is completed. The message, shown in Example 4-16, is repeated the next several times that the target system is rebooted.

Example 4-16 Notice of Configuration File Changes

```
Software installation has installed new configuration files
and saved the previous version in some cases. You may need
to update or merge old configuration files with the newer
version. Please see "Updating Configuration Files" in the
versions(1M) manual page for details.
```

Merging Configuration Files

To avoid compatibility problems, plan to merge configuration files if new versions were created. Use the procedure below to merge configuration files.

1. Identify changed configuration files with this command:

```
# versions changed
```

Example 4-17 illustrates output of the previous *versions* command. (You can use an equivalent command from the Inst application: *admin config changed*.)

Example 4-17 Listing of Changed Configuration Files

```
Configuration Files
m      = modified since initial installation
?      = modification unknown
blank = file is as originally installed
/etc/halt
? /etc/halt.O
m /etc/passwd
  /etc/passwd.N
  /etc/reboot
? /etc/reboot.O
m /etc/services
  /etc/services.N
```

The *.O* versions of the configuration files are the earlier versions. In this case, the no-suffix version contains changes that are required for compatibility with the rest of the newly installed software, that increase functioning, or that fix bugs.

The *.N* versions of the configuration files are the versions created during the installation. They contain changes or new features that can be added to the no-suffix version of the configuration file at your option.

Note: The release notes might have information about the changes that were made to configuration files in the new release.

2. Merge information from both versions of the configuration files.

Use *diff* or *xdiff* to compare *.O* and *.N* files with their no-suffix counterparts. Transfer information that is needed from the *.O* version to the no-suffix version. Add information from the *.N* version to the no-suffix version if you want it.

3. Delete the *.O* and *.N* versions of the configuration files.

If you want to keep *.O* and *.N* files, rename them; otherwise they might be removed automatically during the next installation. When you remove all *.O* and *.N* configuration files, no message about configuration files appears when you reboot the system and the startup process is faster.

Using Supplementary Menus

In some installation situations, it will be useful or even necessary to modify Inst operating parameters or change conditions on the target system to facilitate installation. This chapter describes how to use selections on the View Commands menu and the Administrative Commands menu to make these modifications. It also explains how to enter commands that are not offered on the active menu.

This chapter contains these sections:

- “Giving Commands on Other Menus” on page 63
- “Using the View Commands Menu” on page 64
- “Using the Administrative Commands Menu” on page 70
- “Using the Interrupt Menu” on page 79

Giving Commands on Other Menus

From the Inst Main menu, you can execute commands on the View Commands menu and the Administrative Commands menu without switching menus. To do this, begin your entry with either the *view* or the *admin* command, followed by the selection from the supplementary menu that you want to execute. For example,

```
Inst> admin date
Wed Feb 24 22:58:53 PDT 1993
```

```
Inst>
```

When the View or Administrative Commands menu is active, you can enter Main menu commands without switching menus. To enter a Main menu command from either the View or the Administrative Commands menu, begin your entry with the *return* command, followed by the command from the Main menu that you want to execute. For example,

```
View> return remove ftn.man.relnotes
```

Using the View Commands Menu

The View Commands menu allows you to change the default behavior of the *list* and *step* commands to adjust their output. By default, *list* and *step* post an alphabetic display of all software in the distribution. The size column reports a *delta size*, which is the difference in disk space requirements between the installed software and the software on the distribution. Items on the View Commands menu offer alternatives to these defaults.

To display the View Commands menu, enter the *view* command from the Main menu, as shown in Figure 5-1.

```
Inst> view
```

```
View Commands Menu
```

1. distribution	Show only products on distribution
2. target	Show only products on target
3. filter [setting]	Show or hide products by attribute(s)
4. level [1,2,3]	View (1) product, (2) image or (3) subsystem
5. sort [size,name]	Select sorting of product list
6. sizes	Show total product sizes instead of deltas
7. deltas	Show disk space deltas instead of total sizes
8. filesystems [on,off,fs ..]	Show size/delta breakdown by filesystem
9. show	Show the current view settings
10. return	Go to the Inst Main Menu

```
View>
```

Figure 5-1 The View Menu

Note: The *set* command is a hidden selection on the View Commands menu; you can use *set* whenever the View Commands menu is posted to display and change Inst preferences. (See “Displaying Hidden Commands” on page 120 for more information on hidden commands and Chapter 8 for information on setting preferences.)

For instructions on using commands from other menus while the View Commands menu is active, see “Giving Commands on Other Menus” on page 63.

The target and distribution Commands

The *target* and *distribution* items on the View Commands menu change the view between the distribution inventory and the target. When you change views, a message is displayed to notify you of the new view:

```
View> target
View: target

View> distribution
View: distribution
```

Any *list* or *step* command that you enter after changing the view contains software items from the specified location. Example 5-1 illustrates the effects on *list* output when the view is set to the target. In this example, the installed software on the target does not include patches.

Example 5-1 A list Display From the Target View

```
View> target
View> return list
View: target
Status: I=installed, R=removed, ' '=not installed
Selection: i=install, r=remove, k=keep, u=upgrade

Subsystem Type(s)[bdrc]: b=reBoot needed, d=Default, r=Required,
                        c=client only
u I 4DDN.sw.4DDN [bd]          2854 4DDN Software
u I 4Dwm.sw.4Dwm [d]          3 Desktop Window Manager
u I ViewKit_eoe.sw.base [d]   0 ViewKit Execution Environment, 1.1
```

Example 5-2 illustrates the effects on *step* output when you reset the view to the distribution. Notice that this distribution contains patches.

Example 5-2 A step Display From the Distribution View

```
View> distribution
View> return step
View:      distribution
Status:    N=new, U=upgrade, S=same, D=downgrade, P=patch upgrade
Patches:   A=installable patch, X=uninstallable (missing base product)
Selection: i=install, r=remove, k=keep

Subsystem Type(s) [bdrpc]: b=reBoot needed, d=Default, r=Required, p=Patch
                        c=Client only
Step commands: i,r,k,n,p,... Use ? for more step help.
i N 4DDN.man.4DDN [d] 146 4DDN Manual Pages
```

The filter Command

By default, the output of *list* and *step* contains all subsystems that are available in the distribution inventory; when the view is set to the target, *list* and *step* output contains all software that is either installed or recorded in the installation history. The *filter* command on the View Commands menu alters the output of *list* and *step* by toggling a specified subset of the display off and on.

You can use any status designation, such as *new* (on distribution listings) or *installed* (on target listings), as the filter. You can also use the keywords *all*, *notinstalled*, and *rejected* as filters (see Inst online help for a complete description of filters).

For example, this *filter* command causes removed subsystems to be omitted from *list* and *step* displays of target software:

```
View> filter removed off
```


The command sequence in Example 5-3 illustrates a target listing before and after the previous *filter* command is entered (the *list* legend is omitted from this example):

Example 5-3 Effects of Using View Filters

```
View> .. list

      4DLT                818 4DLT software 3.1
R ClearCase              40653 CASEVision/ClearCase 2.0.1 BL1 (IRIX 5.3)
R FDDIXPress             1012 FDDIXPress, 3.6
  InPerson               8273 InPerson Desktop Conferencing, 1.1
R ViewKit_dev            0 ViewKit Development Environment, Version 1.1
N cosmo                  246 Cosmo Compress Execution Environment 1.1
N dba                    0 Database Accelerator, 5.3
  media_dev              0 IRIS Digital Media Development Environment

View> filter removed off
View> return list

      4DLT                818 4DLT software 3.1
  InPerson               8273 InPerson Desktop Conferencing, 1.1
N cosmo                  246 Cosmo Compress Execution Environment 1.1
N dba                    0 Database Accelerator, 5.3
  dmedia_dev            0 IRIS Digital Media Development Environment
```

The level Command

By default, the *list* and *step* commands display the subsystem components of software on the distribution or target. Using the *level* command from the View Commands menu, you can change the output to a different component in the product hierarchy. The *level* command takes *product* (or *1*), *image* (or *2*), and *subsystem* (or *3*) as arguments.

Example 5-4 illustrates the effects of a *level* command on the first two products shown in Example 5-3 (the *list* legend is omitted from this example).

Example 5-4 Effects of level on a list Display

```
View> level image
View> return list

i N 4DDN.man             146+ 4DDN Documentation
i N 4DLT.sw              794+ 4DLT software3.1
i N InPerson.books      0 InPersonBooks
i N InPerson.man        2215+ InPersonDocumentation
i N InPerson.sw         5294+ InPersonSoftware
```

The sort Command

By default, *list* and *step* sort software alphabetically by product, image, or subsystem name. Using *sort*, you can switch between alphabetic listings of software and listings that are ordered by size.

Example 5-5 shows how the alphabetical listing in Example 5-4 is affected when the *sort* command is used to arrange by size (the *list* legend is omitted from this example).

Example 5-5 Effects of sort on a list Display

```
View> sort size
View> return
Inst> list

i N InPerson.sw          5294+ InPersonSoftware
i N InPerson.books      2172+ InPersonBooks
i N 4DLT.sw             794+ 4DLT software3.1
i N 4DDN.man           146+ 4DDN Documentation
i N InPerson.man        43+ InPersonDocumentation
```

Note: When the *sort size* command is in effect, the ordering of a list is affected by the *sizes*, *deltas*, and *filesystems* command (see “The filesystems Command” below).

The sizes and deltas Commands

By default, *inst* runs with the *deltas* command in effect; that is, *list* and *step* commands report the size of a product or its components as the difference (*delta*) in space requirements between the installed software on the target and the corresponding software in the distribution. You can display the absolute sizes of software, rather than the delta sizes, by using the *sizes* command on the View Commands menu. The *sizes* command remains in effect until you enter a *deltas* command.

Notice in Example 5-5 that a plus sign (+) follows each product size. The plus sign indicates that these products, which in this case are new, will require additional disk space on the target. At times, upgrade products contain a minus sign (-), indicating that the distribution version is smaller than the corresponding target version.

When the view is set to target software, delta sizes represent the change in disk space that occurs when the software is removed or replaced. In this case, the target contains a larger version of the software than the distribution.

The filesystems Command

By default, *list* and *step* commands display all writable filesystems that are local to the target; disk space estimates reflect an aggregate of the software that is selected to be installed on all locally mounted filesystems. Using the *filesystems* command, you can change *list* and *step* output to report space requirements on individual filesystems, on read-only filesystems, and on NFS filesystems that are writable.

Most arguments to the *filesystems* command are paired:

- on, off* toggles space estimates to individual or aggregate filesystems
- local, nfs* specifies whether space estimates reflect locally mounted filesystems (the default) or writable NFS filesystems
- readonly, all* specifies whether space estimates reflect just read-only filesystems or writable and read-only filesystems

filesystem_name any filesystem specification

Any *filesystems* command entry overrides the previous *filesystems* entry.

Example 5-6 illustrates how to display individual filesystems in a list display. Notice that the *sort* command from Example 5-5 is still in effect, so images are sorted by size first on the root filesystem (/), then on */disk_2*, and finally on */disk_3*.

Example 5-6 Displaying Individual Filesystems

```
View> filesystems on
View> return list
```

		/	/disk_2	/disk_3
N	InPerson.sw	5294	0	0
N	InPerson.books	217	0	0
i N	4DLT.sw	794	0	0
i N	4DDN.man	146	0	0
i N	InPerson.man	43	0	0
i N	WorkShop.sw	0	46927	0
S	c++_dev.sw	0	4173	0
	Current free space	94358	487872	169735
-	Selections net change	8449+	51100	0
-	Temporary inst overhead	184+	328	0
=	Minimum free during install	85725	436444	169735
	Final projected free space	85909	436772	169735

Example 5-7 shows how you can change the columns that are shown in list and step displays. Notice that */disk_2* software appears before */software*, as specified by the order of *filesystems* command arguments, and that the */disk_3* filesystem does not appear in the listing because it is not specified as an argument. The *filesystems* command is abbreviated in this example.

Example 5-7 Changing Column Order in list Displays

```
View> fs /disk2 /
View> return list
```

		/disk2	/
i N	WorkShop.sw	46927	0
S	c++_dev.sw	4173	0
N	InPerson.sw	0	5294
N	InPerson.books	0	2172
i N	4DLT.sw	0	794
i N	4DDN.man	0	146
i N	InPerson.man	0	43
	Current free space	487872	48787
-	Selections net change	51100	8449+
-	Temporary inst overhead	328	184+
=	Minimum free during install	436444	85725
	Final projected free space	436772	85909

Using the Administrative Commands Menu

The Administrative Commands menu extends your control over the installation process and over conditions on the target system by allowing you to take these actions:

- Set the preferences that tailor Inst and Software Manager behavior
- Execute specialized installation commands
- Execute IRIX commands

To display the Administrative Commands menu, shown in Figure 5-2, enter this command from the Main menu:

```
Inst> admin
```

Administrative Commands Menu

1. set [preferences]	List all preferences or set/clear a preference
2. date [args]	Display or set the system date
3. files [names]	List files in subsystems
4. space	List disk usage information
5. recalculate	Recalculate space required for installation
6. sh [cmd]	Escape to a shell or run a command
7. shroot [cmd]	Escape to a chrooted shell or run a command
8. relocate [prod [disk]]	Relocate product to a different disk
9. mount [fsname] [dir]	Show mounted filesystems or mount new ones
10. umount [fsname]	Unmount a filesystem
11. mkfs [blockdevice]	Make a new filesystem
12. sethostname	Set name of current host, for networking
13. sethostipaddr	Set host ipaddr - Internet address
14. save filename	Save selection to file
15. load filename	Load selection from file
16. rqs	Perform rqs(1) processing
17. config [changed]	List all or modified config files
18. hardware	List machine-specific hardware information
19. update keepfile	Add N(ew), unselected subsystems to keepfile
20. return	Go to the Inst Main Menu

Figure 5-2 The Administrative Commands Menu

Information on Administrative commands is also available in Appendix A, “Inst Quick Reference,” and from the Inst *help* command.

The set Command

The *set* command controls inst(1M) operating parameters known as preferences. Refer to Chapter 8, “Tailoring Installations,” for a complete discussion of setting preferences.

The date Command

The *date* command on the Administrative Command menu is identical to *date* (see the date (1) reference page), which time-stamps system events. The correct date and time of installations is important because portions of the installation process use *make* (see the date (1) reference page), which relies on dates to determine whether certain commands should be executed. The correct date and time ensure that actions that are sensitive to file dates are performed correctly.

Use this command to display the date:

```
Admin> date
```

Use this command to set the date:

```
Admin> date mmddhhmmyy
```

The *mmddhhmmyy* argument allows two characters each for year (optional), month, day, hour (on the 24-hour clock), and minutes.

The files Command

When you are undecided about installing a subsystem, looking at its contents can help you decide. The *files* command on the Administrative Commands menu displays the names of the files and directories that a subsystem contains. Arguments to the *files* command are one or more subsystem, image, or product name.

Example 5-8 shows the format and output of the *files* command.

Example 5-8 Sample files Display

```
Admin> files eoe.sw.quotas
i U eoe.sw.quotas      1 f etc/config/quotacheck
i U eoe.sw.quotas      1 f etc/config/quotas
i U eoe.sw.quotas      f etc/init.d/quotas
i U eoe.sw.quotas      1 l etc/rc2.d/S10quotas
i U eoe.sw.quotas     193 f usr/bsd/quota
i U eoe.sw.quotas     145 f usr/etc/edquota
i U eoe.sw.quotas     137 f usr/etc/quot
```

The first, second, and third columns are identical to the first three columns of *list* output. The installation status in the second column applies to the entire subsystem; it does not indicate whether an individual file is currently installed.

The fourth column shows the net change in disk space that the file will require if it is installed (the first column contains the letter *i* or is blank) or removed (the first column contains the letter *r*). The *files* list does not include the components of a currently installed subsystem that are not included in the new software, because these files are obsolete. For this reason, numbers in this column are not necessarily equivalent to the net disk space change shown by the *list* command.

The fifth column is a single character that tells the type of file:

<i>f</i>	plain file
<i>d</i>	directory
<i>b</i>	block special file
<i>c</i>	character special file
<i>l</i>	symbolic link
<i>p</i>	fifo (named pipe)

The sixth column displays the name of every file in every subsystem that matches the arguments specified in the *files* command.

To get a list of the files in an installed subsystem, use the *view* command on the Main menu to set the view to the target system. Then return to the Administrative Commands menu and enter this command:

```
Admin> files names
```

The space Command

Disk space summaries are displayed at the end of the output from *list* and *step* commands. You can request a disk space summary at other times in the session by issuing the *space* command, as shown in this example:

```
Admin> space
Disk space summary (Kbytes):    /
Current free space              426132
- Selections net change         2853+
- Temporary inst overhead      2+
= Minimum free during install  424867
Final projected free space      424967
```

The *space* command does not force a recalculation of disk space (see the discussion of *recalculate*, which follows), so if you enter the space command before space calculations are performed, the net change amount is shown as 0 blocks, even if you have made installation or removal selections.

The output of *space* is affected by the *filesystems* command on the View Commands menu (see “The filesystems Command” on page 69) and by the *delayspacecheck* preference (see “Delaying Disk Space Checking” on page 122). For more information on space checking, see “Controlling Disk Space Checks and Reports” on page 121.

The recalculate Command

If you use *sh* or *shroot* from Inst to modify files on the target system, Inst automatically updates disk space calculations. You can also enter the *recalculate* command from the Administrative Commands menu to request a new space check after making file modifications:

```
Admin> recalculate
```

This command always performs a detailed space check, even when the *detailspacecheck* preference is off (see “Changing Detailed Space Checking” on page 122). The output of the *recalculate* command is the same as the output of the *space* command (above). See “Controlling Disk Space Checks and Reports” on page 121 for more information.

The `sh` and `shroot` Commands

During an Inst session, it is possible to invoke a shell, enter shell commands, and return to Inst without ending the session. This procedure is useful for removing user files to free disk space. Use either of these commands to invoke a shell from the Administrative Commands menu:

```
Admin> sh
Admin> shroot
```

The `sh` command invokes a regular shell, and the `shroot` command invokes a `chroot(1M)` shell. By default, the shell is specified by the SHELL environment variable and is usually set to `/bin/sh` or `/bin/csh`. To invoke a different shell, use the `set` command to specify it:

```
Admin> set shell shell_path
```

Note: When you use the `sh` command during a miniroot installation, remember that the root and user filesystems are mounted as `/root` and `/root/user`. But if you use `shroot`, the root is changed to `/root` and pathnames are unnecessary for most commands. For this reason, using `shroot` is usually preferable to using `sh`.

Miniroot installations impose some limits on what can be done with `sh` and `shroot`. Because the miniroot supports only a subset of IRIX commands, commands in `/root` and `/root/usr` might be unavailable or incompatible with the miniroot kernel. If you are installing from a local distribution source, you cannot use `sh` or `shroot` to access remote systems. If you are using a remote distribution source, you can access the distribution server.

The `relocate` Command

Normally, the installation utility installs software in predesignated directories. The `relocate` command allows you to install a product in a different directory whose name you specify. The `product` argument must specify a product-level component (see Figure 1-1 on page 6). The `dir` argument can be any IRIX pathname, including the mount point of a disk (`/d2`, for example). The `relocate` command is especially useful for installing software on supplementary disks.

The mount and umount Commands

You can identify mounted filesystems by entering the *mount* (see the *mount (1M)* reference page) command with no arguments:

```
Admin> mount
```

If you are performing a miniroot installation, you can mount additional filesystems with this command:

```
Admin> mount filesystem mount_directory
```

Remember that, during a miniroot installation, filesystems are mounted at */root* rather than at */*.

To unmount filesystems, you must be performing a miniroot installation. The *umount(1M)* command has this syntax:

```
Admin> umount filesystem
```

The mkfs Command

During a miniroot installation, you can use the *mkfs* command to create an EFS or XFS filesystem. If you enter *mkfs* without arguments, the root filesystem (*/*), and the user filesystem (*/usr*), if it is present, are erased and re-created. After your entry, you are prompted to choose the filesystem type; if you choose an XFS filesystem, the block size for the new filesystem is set to 4,096 bytes per block.

Caution: This command erases the entire contents of these filesystems; do not use it if the target filesystem contains information that must be preserved.

Re-creating the root filesystem (and */usr*, if it is present) has these effects on an installation:

- All local information in configuration files is lost.
- All information in the installation history database is lost.
- If you are using a remote installation server, you are required to enter the server's IP address when specifying the distribution source.

When you use this command to make filesystems other than root (/) and /usr, you must specify the device special filename where the filesystem resides. For example, this command creates a filesystem on partition 0 of SCSI disk 2:

```
Admin> mkfs /dev/dsk/dks0d2s0
```

If you are creating an XFS filesystem, *mkfs* prompts you to choose the block size (512 or 4096) for the new filesystem. (See the *mkfs(1M)* and *intro(7M)* reference pages for more information on specifying devices; see *IRIX Admin: Disks and Filesystems* for detailed information on creating filesystems.)

To access the newly created filesystem, *device*, create a directory mount point and mount the filesystem:

```
Admin> shroot  
# mkdir mount_directory  
# exit  
Admin> mount device mount_directory
```

The *sethostname* and *sethostipaddr* Commands

The *sethostname* (equivalent to *hostname*) and *sethostipaddr* (equivalent to *hostid*) commands set the network name and address of the target system; either of these commands restarts networking. You can only execute *sethostname* and *sethostipaddr* during a miniroot installation.

The *save* and *load* Commands

The *save* command creates a selections file that contains the distribution location and the selections that are in effect when you enter the *save* command. In subsequent sessions, you can specify the selections file as an argument to the *load* command to automate the selections process. It is also possible to use the selections file that you saved to install software on other systems: the argument to a *load* command can include a remote host specification.

By default, selections files are saved in the directory in which you invoked *Inst*, but you can change the directory location by specifying an alternate directory in the filename specification (see the *-F* option of *inst(1M)*).

This sample entry creates a selections file in */usr/tmp* called *select.sav1*:

```
Admin> save /usr/tmp/select.sav1
Save selections to file: /usr/tmp/select.sav1
```

Later in the session, or in a subsequent installation session, this command retrieves the file and applies the selections that it contains:

```
Admin> load /usr/tmp/select.sav1
```

This command, issued during an installation on a different target host, retrieves *select.sav1* so that it can be used to install software on the target:

```
Admin> load sparkplug:/usr/tmp/select.sav1
```

The format of *select.sav1* is illustrated in Example 5-9. The first two lines identify the selections file to the Desktop. When users double-click the selections file icon, Software Manager starts, using the selection file. The remaining lines identify the distribution location and the selections to be made (an internal version number appears at the end of each selection item).

Example 5-9 Sample Selections File

```
#!/usr/sbin/SoftwareManager -F
#Tag 000109B1
from easyboy:/dl/IRIX_6.3/dist

k license_eoe.man.license_eoe 1022130500
r license_eoe.man.relnotes 1022130500
i license_eoe.sw.license_eoe 1022130500
r netls_nodelock.man.relnotes 1022130500
i netls_nodelock.sw.netls_nodelock 1022130500
i netls_root.sw.hdr 1022130500
i netls_root.sw.lib 1022130500
```

The config Command

The *config* command posts a list of configuration files that are present on the target. If there are differences in a newly installed configuration file and the previously installed version, an *m* is posted to the left of the filename in *config* output. If the previous version of a configuration was made obsolete by the installation, the obsolete version is listed with a *.O* suffix. If a newly installed configuration file is available but not required by the new software, the newly installed version file is listed with a *.N* suffix. (See “Managing Configuration Files” on page 60 for details.)

The hardware Command

The *hardware* command lists information about target system components. This example shows *hardware* output for an Indy workstation target:

```
Admin> hardware
Hardware:
CPUARCH=MIPS2
CPUARCH=R4000
IP22
CPUBOARD=IP22
MODE=32bit
SUBGR=NG1
GFXBOARD=NEWPORT
```

The updatekeepfile Command

The file *\$rbase/var/inst/.keepfile* is a user-created text file that suppresses automatic product selections during initialization (see “Automatic Initialization Functions” on page 42). Any new product that you list in the *keepfile* is not marked for default installation during the automatic selection process. Only new products can be listed in the *keepfile*; automatic selections of updated products cannot be suppressed with this file.

Using the Interrupt Menu

It might sometimes be necessary to interrupt Inst during a processing operation, such as reading the distribution or installing software selections. When an interruption is necessary, press Ctrl+C to suspend processing and display the Interrupt menu.

Figure 5-3 illustrates the Interrupt menu.

```
Error/Interrupt Menu

1. stop                Terminate current command
2. continue           Continue current command
3. set [preferences]  List all preferences or set/clear a preferenc
4. help [topic]       Get help in general or on a specific word
5. sh [cmd]           Escape to a shell or run a command
6. shroot [cmd]       Escape to a chrooted shell or run a command

Interrupt>
```

Figure 5-3 The Interrupt Menu

Retrying and Operation

Under certain error conditions, Inst displays the Interrupt Menu automatically. When it is displayed automatically, the Interrupt Menu frequently contains a “retry” option (see Example 4-13), which repeats the operation that caused the failure. Because a common source of the failure is a network timeout, the repeat attempt is likely to be successful.

If you see a “retry” option, you can safely assume that you can select it without compromising the installation. Check the ERROR line above the Interrupt Menu for insights into the cause of the failure; then take the appropriate corrective action (if any) before proceeding.

Stopping an Installation

The *stop* command cancels the command in progress and suspends the installation. If you use *stop* while Inst is reading the installation history or distribution inventory, or while it is calculating dependencies, Inst discards the processing results. For this reason, Inst must repeat the readings or calculations before executing subsequent *list* commands, selection requests, or the *go* command.

If you use *stop* while Inst is installing and removing software, it makes a record of pending selections. At the next *go* command, Inst restarts the installation process on any uncompleted product installations; then it continues processing the selections that were pending.

Caution: Using *stop* during installation can leave software in an inconsistent state that requires removal and reinstallation of the affected product.

To resume the installation after issuing a *stop* command, use the *continue* command. The *continue* command resumes processing from the point at which it was interrupted.

Using Other Interrupt Menu Commands

The *set*, *help*, *sh*, and *shroot* commands on the Interrupt menu are identical to those on the Administrative Commands menu. See “A Closer Look at Preferences” on page 126 for information on using the *set* command; see Table A-1 on page 182 for a list of topics covered by the *help* command; and see “The *sh* and *shroot* Commands” on page 75 and “Interrupt Menu Commands” in Appendix A on page 192 for details on using the *sh* and *shroot* commands.

Inst for Experts

This chapter provides instructions to users who are already familiar with the basic operation of Inst. It contains cursory installation instructions as an alternative to the detailed instructions in Chapter 5 and elaborates options for using Main menu commands. Finally, this chapter explains how to use features on the other Inst menus.

This chapter contains these sections:

- “Alternative Ways to Specify a Distribution” on page 83
- “Installing Software Noninteractively” on page 85
- “Using Command Shortcuts” on page 85
- “Using step to Make Selections” on page 90

Alternative Ways to Specify a Distribution

The *from* command specifies the distribution source for an installation session or portion of the session (see “Step 2: Specifying the Source” in Chapter 4 for basic information). This section suggests several alternatives to the specification methods described in Chapter 5. “Using a Distribution Keyword” on page 90 contains additional information.

Using the \$dist Variable

The *\$dist* variable is a shorthand notation to specify the location of a distribution. The value of *\$dist* is the current value of the *dist* preference (see “A Closer Look at Preferences” on page 126). The *dist* preference is automatically set each time Inst reads product descriptions from a distribution source. When Inst is invoked, the value of *dist* is set to the last software distribution source that was specified in an Inst session.

Use *\$dist* as an argument to *from* to specify the location of the distribution:

```
Inst> from $dist
```

You can check the value of *dist* at any time by entering this command:

```
Inst> view set dist
```

Specifying One Product as the Source

A distribution specification can contain the name of a specific product to be installed. These sample entries illustrate how to specify a single product as a distribution source:

```
Inst> from server:/CDROM/dist/product_name
Inst> from $dist/product_name
```

When you specify a product name, only the specified product can be installed. However, this method reduces the time required for disk space checking and verification of the installation history.

Specifying an Alternate User Account

As a default, Inst uses the *guest* account to connect to an installation server (see “Configuring an Installation Account” on page 18). To use an alternate account on the server, specify the user account name in your distribution specification:

```
Inst> from user@server:pathname
```

Specifying the Source on the Command Line

For a live installation, you can use the Inst command-line argument *-f* to specify the distribution source. This is equivalent to giving the *from* command on the Inst Main menu. These examples illustrate *inst -f* command entries:

```
# inst -f /CDROM/dist
# inst -f server:pathname
# inst -f server:pathname/product
```

In your entry, specify the pathname of the software distribution for *source* (*/CDROM/dist*, for example); if the distribution is on a remote server, include the server name in your specification (*easyboy:/CDROM/dist*, for example). If you wish to install an individual product, substitute the shortname of the product for *product* in your entry (*easyboy:/CDROM/dist/InPerson*, for example). See “Step 2: Specifying the Source” on page 41 for additional information on specifying a distribution source.

Installing Software Noninteractively

It is possible to install software without using Inst menus; however, only live installations can be performed noninteractively. Noninteractive installations install the subsystems that are preselected by Inst (see “Required and Default Subsystems” on page 7 for information on these selections). In cases where the distribution contains an update to installed software, the older version on the target is replaced by the update. In cases where the distribution contains software that is not installed on the target, the default subsystems are installed (these are marked with *d* in *list* displays).

To install software noninteractively, use the *-a* argument (automatic) on the *inst* command line. To specify the location of the software distribution in the automatic installation, use the *-a* argument with the *-f* argument in your command. If you do not use the *-f* argument, the default is the software distribution source that you specified when you last used Inst.

In Example 6-1, the *ftn_dev* product is installed noninteractively from a software distribution directory on a remote workstation.

Example 6-1 Sample of a Noninteractive Installation

```
% /bin/su -  
Password:  
# inst -a -f easyboy:/d/IRIX_5.3/ftn_dev  
#
```

See also the optional RoboInst software described in Chapter 9, “Automating Installations With RoboInst.”

Using Command Shortcuts

The discussions that follow explain how to save keystrokes when using Inst commands. These shortcuts apply to all Inst menus.

Giving Commands by Number

Instead of command names, you can give command numbers at the prompt. For example, this command is the same as entering the *remove* command:

```
Inst> 7
```

Abbreviating Command Names

Inst recognizes the shortest unique abbreviation of any command, so your command entry can be brief. You can also use command abbreviations in combination with keywords (see “Using Keywords as Arguments” on page 87). The sample entry below, which is an abbreviation for the *list* command used with the keyword *downgraded*, is the same as entering *list downgraded*:

```
Inst> l D
```

Using Wildcards in Subsystem Names

When you enter subsystem names as arguments to Inst commands, you can use wildcards to shorten your entries. Inst accepts these shell-style wildcards in subsystem names:

- ? Matches one character.
- * Matches any combination of characters but applies only to the product, image, or subsystem portion of the name in which it is used. In other words, the asterisk (*) does not match characters in the entire subsystem name; it matches only the characters that appear in one segment of a subsystem name (see “Using Product and Image Names” on page 87 for an explanation of name segments).
- [] Matches any enclosed characters or a range of characters separated by a dash.

Table 6-1 illustrates the use of wildcards in product names.

Table 6-1 Subsystem Names Specified With Wildcard

Entry	Meaning
<code>eoe.sw.*</code>	All <i>sw</i> subsystems in the <i>eoe</i> product
<code>*.man.*</code>	All <i>man</i> subsystems in the distribution
<code>*.man</code>	All <i>man</i> images in the distribution
<code>eoe*</code>	All products whose names begin with <i>eoe</i>
<code>[a-c]*.sw.*</code>	All <i>sw</i> subsystems in products whose names begin with a, b, or c

Using Product and Image Names

Any command that accepts subsystems names as arguments also accepts product and image names as arguments. This means that the trailing asterisk wildcards (*.*) are not necessary to specify all subsystems in an image or product.

Example 6-2 illustrates commands that use product and image names as arguments.

Example 6-2 Commands With Product and Image Arguments

```
Inst> list eoe
Inst> install *.man
Inst> keep *
```

You can tell whether you are specifying a subsystem, image, or product name by the number of segments in the name. Product names contain one segment and no dots, image names contain two segments separated by one dot, and subsystem names contain three segments separated by two dots. These rules are true even when you use wildcards.

Example 6-3 illustrates the structure of names in the product hierarchy.

Example 6-3 Sample Product, Image, and Subsystem Names

```
dmedia_dev
dmedia_dev.sw
dmedia_dev.sw.audio
```

Using Keywords as Arguments

The *list*, *install*, *remove*, *keep*, and *step* commands take arguments called *keywords*. Keywords are shorthand designations for groups of subsystems, images, or products. Many keywords have a long form and a single-letter abbreviation.

Table 6-2 lists each keyword, its abbreviation (if it has one), and a description of the subsystems, images, or products that the keyword designates.

Table 6-2 Keywords

Keyword	Abbrev.	Meaning
all	a	All subsystems in <i>showprods -a</i> output plus all subsystems in the software distribution.
clientonly	c	All subsystems to be installed only on NFS client systems.
conflicting	C	Subsystems that are generating conflicts
default	d	All subsystems that are recommended for installation by the manufacturer. The letter d appears after the subsystem name in <i>list</i> output.
<dist>:		All subsystems in all distributions whose name contains the string specified by <i>dist</i> . In the case of a CD, the string specifies the CD title.
distribution	dist	All subsystems in the distribution.
downgrade	D	All subsystems for which the version in the distribution is a downgrade of (older than) the installed version on the target. The letter D appears in the second column of <i>list</i> output
feature	(none)	Feature stream of release. Includes content of maintenance stream (see keyword "maint") plus new software features.
hardware	h	All subsystems containing hardware-specific files.
install	i	All subsystems that are selected for installation. The letter i appears in the first column of <i>list</i> output
installable	A	All subsystems in the software distribution except the patches, machine-dependent subsystems, and client-only subsystems that do not apply to the target.
installed	I	All subsystems for which the installed target version is the same as the distribution version. The letter I appears in the second column of <i>list</i> output.
keep	k	All installed subsystems that are not selected for reinstallation, replacement, or removal.

Table 6-2 (continued) Keywords

Keyword	Abbrev.	Meaning
maint	(none)	Maintenance stream of release. Includes bug fixes essential to system stability and compatibility, plus support for new hardware and hardware upgrades.
new	N or n	All subsystems in the distribution that are not installed on the target and are not upgrades or downgrades of an installed subsystem. The letter N appears in the second column of <i>list</i> output.
notinstalled	(none)	All subsystems that were never installed on or removed from the target.
patches	(none)	All patches in the distribution.
patchupgrade	P	All patches on the distribution that are upgrades to installed patches. The letter P appears in the first column of <i>list</i> output.
prereq	(none)	All prerequisites for selected software. Automatically resolves conflicts.
removable	L	All installed subsystems that can be selected for removal.
remove	r	All subsystems that are selected for removal. The letter r appears in the first column of <i>list</i> output.
removed	R	All subsystems that were once installed on the target and later removed. The letter R appears in the first column of <i>list</i> output.
required	q	All subsystems that are required for proper operation of the target. The letter r appears after the subsystem name in <i>list</i> output.
same	S or s	All subsystems for which the version on the distribution and the version in the target are the same. The letter S appears in the second column of <i>list</i> output.
stale	(none)	All installed subsystems that must be reinstalled because of hardware changes.
standard	(none)	All upgraded subsystems and any subsystems that are new to the product and designated as default.
target	targ	All subsystems on the target.

Table 6-2 (continued) Keywords

Keyword	Abbrev.	Meaning
upgrade	U	All subsystems for which the distribution contains a version that is newer than the installed version on the target. The letter U appears in the second column of <i>list</i> output.
upgraded	u	All subsystems on the target that are being replaced by a distribution version (usually newer) as part of the installation of a related product.

Using a Distribution Keyword

Inst interprets any argument to a *list* or selection command that ends in a colon (:) as a *distribution keyword*; that is, the argument is considered a distribution specification. A distribution keyword is useful when multiple distributions are open. You can use distribution keywords with other keywords in your entries. For example, this command installs all default subsystems from an open distribution on server cameo:

```
Inst> install cameo:/6.3/options: d
```

Using step to Make Selections

The *step* command allows you to display the distribution or target inventory, one item at a time, as you make selections. By default, *step* displays subsystems in the distribution inventory, but, like *list*, the output of *step* is determined by the current *view* settings. The *step* command takes the same keyword arguments and wildcards as *list*.

The items that *step* displays are determined by the arguments in your entry. For example, this *step* command causes only upgraded products to be displayed:

```
Inst> step U
```

After the entry, Inst posts selection information, followed by the first upgraded product in the distribution, as shown in Example 6-4.

Example 6-4 A step Command Display

```

Current View:
  Location:  distribution
  Status:   N=New,U=Upgrade,P=Patch upgrd,S=Same,D=downgrade,, ' '=Not Installed
  Selection: i=install, r=remove, k=keep
  Level:    subsystem
  Name:     short

  Subsystem Type(s) [bdrp]: b=reBoot needed, d=Default, r=Required, p=Patch]

Step commands: i,r,k,n,p,... Use ? for more step help.
i S rebind.man.rfind [d]                0 Rfind Man Pages

```

As each software item is displayed, use the *i* (install), *r* (remove), or *k* (keep) to make selections. If you use a capital *I*, *R*, or *K*, your action applies to all subsystems in the product. To leave an item unchanged, press Enter. After your selection, the next subsystem is displayed for selection:

```

i D U 4dwm [d]                800+ Desktop Window Manager
  D U cadmin                  1831+ Desktop Administration, 5.2

```

Continue until you finish making selections. Type ? at any time for help. You can quit the *step* process at any time by typing q.

Using Advanced step Features

The *step* command allows an interactive operating mode that enables you to make subsystem selections with a special key or key combinations. In some cases, you can request a particular action using several different keys. Some special keys provide searching and listing capabilities.

Table 6-3 lists the special keys that you can use during *step* operations.

Table 6-3 Requests in *step* Operations

Level Affected	Requested Action	Keys
Subsystem	Select this subsystem for installation.	i or u or <right-arrow>
	Select this subsystem for removal.	r
	Keep the current subsystem version; cancel the current request.	k or <left-arrow>
	Select this subsystem for installation if it is marked with a <i>d</i> .	d
	No change to the current request for this subsystem.	<Enter> or n or j or <down-arrow>
	No change to the current request for this subsystem; display the previous subsystem.	p or <up-arrow> or -
	List all previous subsystems.	l
	List the files in this subsystem.	f
Products	Select the remaining subsystems in this product for installation.	I or U
	Select the remaining subsystems in this product for installation if they are marked with a <i>d</i> .	<Shift><right-arrow>
	Select the remaining subsystems in this product for removal.	R
	Keep the current version of this product; cancel the current request for remaining subsystems in the product.	K or <Shift><left-arrow>
	Select subsystems in this product for installation if marked with a <i>d</i> .	D
Retain the current request for all subsystems in this product and display the first subsystem in the next product.	N or J or <Shift><down_arrow>	

Table 6-3 (continued) Requests in *step* Operations

Level Affected	Requested Action	Keys
	Retain the current request for this product and display the first subsystem of the previous product.	P or <Shift><up-arrow>
All levels	Search for <i>pattern</i> and display the product that contains it.	/ <i>pattern</i> <Enter>
	Display help for <i>step</i> operation.	h
	Quit <i>step</i> operation.	q

Maintenance Tips

This chapter contains procedures that may be necessary to keep the software installation at your site up to date. The chapter contains these sections:

- “Installing Maintenance and Feature Releases” on page 95
- “Installing Optional Software Products” on page 99
- “Installing Patches” on page 99
- “Installing Software for Hardware Upgrades” on page 103
- “Installing Reference Pages” on page 104
- “Reinstalling an Older Software Release” on page 105
- “Avoiding Compatibility Problems” on page 108
- “Removing Software” on page 109

Installing Maintenance and Feature Releases

The distribution CD for maintenance and feature releases contains two software streams: the maintenance stream contains bug fixes and basic support for new hardware; and the feature stream contains the maintenance stream software as well as new software features (see “Software Release Types” on page 3). The new files in any maintenance or feature release overwrite the corresponding files in the installed subsystem.

The first time you install an intermediate release, `Inst` prompts you to specify the stream that you wish to install. After you select a stream, `Inst` makes its installation selections from the appropriate stream. In many cases, you can examine the initial selections with *list* and enter the *go* command without making any selection changes.

When Inst makes its initial installation selections on maintenance or feature releases, it automatically selects any updates to existing subsystems, provided that the appropriate version of the subsystem is currently installed. If the required version of a subsystem is not installed, you must upgrade to the required version before installing its replacement in the maintenance or feature release.

Switching Streams

Normally, the previously-chosen stream type is installed on a particular operating system base release. However, you can switch streams. To do this, use the Inst commands *install feature* or *install maintenance* when the Inst prompt first appears. Then, follow the procedures discussed in the following two sections.

Feature-to-Maintenance Switches

If you are currently running a feature release and you use the *install maintenance* command to install the maintenance stream, you must open all CDs from the base release as part of the installation. You then must reinstall the portions of the base release that are affected by the maintenance release that you plan to install. You can determine which files are effected by entering the *conflicts* command.

Maintenance-to-Feature Switches

If you are currently running a maintenance release and you use the *install feature* command to install the feature release, you will be prompted with conflict messages in your installation similar to the following:

```
Overlay product eoe.sw.base (1275066720) cannot be
installed because of missing prerequisites: base
product eoe.sw.base (1274627340)
```

```
1a. Do not install eoe.sw.base (1275066720)
```

```
1b. Also install base product eoe.sw.base
(1274627340) from an additional distribution --
insert another CD or specify another software
distribution.
```

If you see conflict messages like this, open the distribution or CD that contains the missing product, then select the product for installation.

Maintenance and Feature Information in List Displays

In Inst subsystem listings, maintenance and release components are listed by the name of the subsystem to which they apply. The letter `o` to the right of the subsystem indicates that it is an overlay to the subsystem.

Example 7-1 illustrates the legend and a listing for a maintenance or feature release of product X. Notice that two subsystems are designated as installable upgrades (UA) and one subsystem is listed as an uninstallable upgrade (UX).

Example 7-1 Sample Listing for Maintenance or Feature Software

```
Inst> l
View:      distribution
Status:    N=new, U=upgrade, S=same, D=downgrade
Status:    A=installable patch/overlay, X=uninstallable patch/overlay
Selection: i=install, r=remove, k=keep

Subsystem Types [bdro]: b=reBoot needed, d=Default, r=Required, o=overlay
i UA eoe.man.eoe [o]      8+  manpages - 1008000004
i UA eoe.sw.eoe [o]      0   sw - 1008000004
UX eoe.sw.eoe [o]      4-  sw - 1008000004
```

Maintenance and Feature Release Installation Procedure

Use the procedure below as a guideline for installing a maintenance or feature release:

1. Determine whether a miniroot installation is required.

Read the release notes or the CD booklet that accompanied the release to determine whether it requires a miniroot installation.

2. Load the installation CD into the CD-ROM drive.

One CD in a maintenance and feature release contains installation utilities; this CD must be loaded first. Even if you are installing a maintenance or feature release and a base release at the same time, the first CD that you load must be the maintenance or feature release CD containing the installation tools.

3. Invoke Inst from the installation CD.

If the release requires a miniroot installation, load the miniroot from the installation CD (see “Loading From a Local CD” on page 30 if you need instructions). If the release does not require a miniroot installation, invoke `Inst` to start the installation (see “Starting a Live Installation” on page 36).

4. Load the remaining CDs, if there are others.

Load the remaining CDs for the release (see “Opening Multiple Distributions” on page 23).

5. Select products and subsystems for your specific installation. Give the following commands, depending on which stream you wish to install:

- Give the following commands to upgrade a **previously-chosen** maintenance or feature stream:

```
keep *  
install standard
```

This clears any existing selections, selects upgrades to currently installed products, and selects products required by these upgrades. You can continue to customize selections after executing these commands.

- Give the following command to switch to the **maintenance stream** release:

```
install maint
```

This clears any existing selections, selects maintenance stream upgrades and downgrades, selects products required by these upgrades, and sets the release stream preference to maintenance. Also, it turns the *neweroverride* option on. This option allows products viewed as older (that is, all maintenance products) to be selected without generating conflicts. You can continue to customize selections after you execute this command.

- Give the following command to switch to the **feature stream** release:

```
install feature
```

This clears any existing selections, selects feature stream upgrades, selects products required by these upgrades, and sets the release stream preference to maintenance. You can continue to customize selections after you execute this command.

6. Resolve any conflicts.

Enter the *conflicts* command to display any conflicts among the subsystems selected for installation. Resolve the conflicts before launching the installation. If you switched streams (see “Switching Streams” on page 96), the conflicts will include those files in the base release that are affected by the switch.

7. Enter the *go* command to launch the installation.
8. Exit Inst.
9. Correct your configuration files, if necessary.

If you see a message about changed configuration files after the installation is complete, follow the directions in “Merging Configuration Files” on page 61 to merge configuration files. If you do not merge changed configuration files, a message about changed configuration files is posted during the next several system reboots.

Installing Optional Software Products

Software options are products that you may or may not choose to run on a system. They are usually purchased separately from a computer system but may also be shipped as a complimentary offering with a new system or when a new version of a purchased option is released. When you install a software option, keep these points in mind:

- The release notes for a software option explain whether a miniroot installation is required.
- If the option requires a miniroot installation and the distribution CD contains installation tools, use the installation tools on the CD. If the option requires a miniroot installation and the distribution CD does not contain installation tools, use the installation CD from your most recently installed release.

Installing Patches

A patch release contains replacement files that can be installed to modify a particular software product. The subsystems that are affected by a patch may span several products. For example, a patch for the FDDI product might contain replacement files for *oe* and *fddivis* subsystems.

The prerequisites for a patch are the software components in the original version of the affected product; occasionally, a previous patch is a prerequisite to a later patch. In some cases, a patch is incompatible with a previous patch and the older patch must be removed before you install the new one.

When you install a patch, Inst automatically moves the original files before installing their patch replacements. The original files are copied to a directory that Inst creates specifically to save such files, `/var/inst/patchbase`. This makes it possible to remove a patch and replace it with the original version of the product files.

Interpreting Patch Information in list Displays

In Inst subsystem listings, patches are identified by their patch number and product subsystem name. For example, `patchSG0000034.bvo_sw.bvo` identifies a patch to a subsystem of the Broadcast Video Option product.

If patches are present on a distribution, the legend of a *list* display and the software components that *list* posts include patches. Example 7-2 illustrates the list legend and patch entries in a *list* display.

Example 7-2 Sample list Display of Patches

```
Inst> list
View:      distribution
Status:    N=New, U=Upgrade, S=Same, D=Downgrade, P=Patch upgrade
Patches:   A=installable patch, X=uninstallable patch (missing base product)
Selection: i=install, r=remove, k=keep

Subsystem Type(s) [bdrpc]: b=reBoot needed, d=Default, r=Required, p=Patch,
                          c=client only

DA patchSG0000000.eoe1_sw.unix [bp]    0  IRIX Execution Environment
SA patchSG0000001.eoe1_sw.unix [bp]    0  IRIX Execution Environment
PA patchSG0000002.eoe1_sw.unix [bp]    0  IRIX Execution Environment
NA patchSG0000005.4DDN_sw.4DDN [bd]    0  4DDN Software
NX patchSG0000011.dwb_sw.aps [p]       0  Autologic APS-5 Fonts
```

The mnemonic tags on patches have these meanings:

- DA (Downgrade Applicable) This patch applies to an installed base subsystem; it is a downgrade to another patch that is already installed.
- NA (New Applicable) This patch applies to a base subsystem that is installed or available for installation; it does not apply to any installed patch.
- NX (New Uninstallable) This patch applies to software that is not installed and not available for installation from this distribution.

- PA (Patch Upgrade) This patch applies to an installed base subsystem; it upgrades a patch that was already installed.
- SA (Same Applicable) This patch is already installed.

Patch Installation Procedure

Use the procedure below as a guideline for installing a patch release:

1. Read the release notes to decide whether to install the patch.

Typically, the reported errors that result in a patch occur only under certain operating conditions. For this reason, installing a patch is necessary only if a system or site is experiencing a problem that the patch addresses. Review the online release notes that accompanied the patch release to determine whether to install it or not.

2. Invoke Inst (from the miniroot, if necessary).

Check the release notes for installation instructions. Invoke Inst from the miniroot if the patch requires a miniroot installation (see “Starting a Miniroot Installation” on page 30 if you need help); otherwise, invoke Inst from the IRIX command line (see “Starting a Live Installation” on page 36 if you need help).

Note: For live installations, you can use *inst -f* and specify the location of the patch distribution in your command.

3. Select the patch subsystems for installation.

Use the *install* command from the Inst Main menu to select all subsystems in the patch; installing a subset of patch subsystems is not recommended. If the patch release contains a patch set, it is advisable to install the entire set rather than individual patches.

```
Inst> install patchname
```

4. Resolve any conflicts.

Enter the *conflicts* command to display any conflicts among the subsystems selected for installation. Resolve the conflicts before launching the installation.

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

Patches contain no known conflicts with standard installed products; however, conflicts might occur with other patches that you have installed. Use the resolution suggested in the conflicts message to determine your course of action.

6. Exit Inst.

Removing Patches

You can remove a patch after it is installed if you wish. When you remove a patch, the original product files in `/var/inst/patchbase` are automatically reinstalled on the system. Use this procedure as a guideline for removing patches.

1. Invoke Inst (from the miniroot, if necessary).

Check the release notes for the product. If the product requires a miniroot installation, you must invoke Inst from the miniroot to remove the patch (see “Starting a Miniroot Installation” on page 30 if you need help); otherwise, invoke Inst from the IRIX command line (see “Starting a Live Installation” on page 36 if you need help).

2. Select the patch for removal.

Use the `remove` command from the Inst Main menu to remove the patch:

```
Inst> remove patchname
```

Note: Do not remove portions of patches; to preserve the integrity of the software product, remove the entire patch.

3. Enter the `go` command to complete the removal.

Removing Original Product Files

Sometimes it may be necessary to remove original product files from `/var/inst/patchbase` (to save disk space, for example). A special argument to the `versions` command removes the files that a patch has replaced. However, there are important reasons why you should create a backup copy of `/var/inst/patchbase` before removing any of its contents:

- Original product files are not recoverable after they are removed from `/var/inst/patchbase`.
- Original product files are sometimes required for subsequent patches to the product.

Use the `versions` command with the `removehist` argument to remove original product files from `/var/inst/patchbase`.

Caution: Before you use this command, make a backup copy of `/var/inst/patchbase`:

```
# versions removehist patchname
```

Installing Software for Hardware Upgrades

In some cases, changes to software are required when you install a hardware upgrade on a system that is already in service. As a general rule, adding memory, bitplanes, and disks requires no change to the software, but other hardware upgrades require changes to the installed software. The documentation that accompanies the upgrade explains whether a software installation is necessary.

Installing Accompanying Product Releases

Some hardware upgrades are shipped with a software distribution in the form of a software product release. Use the directions in “Installing Optional Software Products” on page 99 to install this type of software distribution. If the software distribution contains installation tools, you must use them.

Reinstalling the Same Software

Some hardware upgrades are not shipped with a software distribution, but they require that you reinstall some software after the hardware upgrade is completed. For example, upgrades to a CPU board or the graphics subsystem require a software reinstallation. In such cases, the reinstallation is necessary because the hardware-specific files that were installed for the original hardware are not appropriate for the new hardware.

When Inst determines that a system contains new hardware, it notifies you that it has selected a new set of hardware-dependent files for installation:

```
Your system hardware configuration has changed since you
last installed software, and some subsystems will therefore
need to be re-installed for proper operation.
```

Use this procedure to reinstall all software:

1. Select all installed software for installation.

Because you are reinstalling the same software, Inst automatically marks previously installed subsystems (including the hardware-specific subsystems that require replacing) with a k (keep). For this reason, you must explicitly select subsystems for installation to force Inst to perform the installation. This command selects all previously installed software for reinstallation:

```
Inst> install stale
```

2. Check the output of *list* after your entry.

Every subsystem in the distribution inventory that is marked with an I (installed) in the second column should also contain an i (install) in the first column.

3. Launch the installation.

```
Inst> go
```

4. Quit Inst.

```
Inst> quit
```

Installing Reference Pages

The reference pages (manual pages) that apply to a product are shipped as a software component of the product. For example, all reference pages and software for Digital Media Development software are shipped in a subsystem of the *dmedia_dev* product. The subsystems containing reference pages consist exclusively of reference pages.

The software subsystems in a product usually have a corresponding reference page subsystem. The names of the software and reference page subsystems differ only in the image segment of the name. The name for a reference page subsystem always contains the letters *.man* in the image segment. For example, *dmedia_dev* contains the software subsystem *dmedia_dev.sw.movie* and a reference page subsystem called *dmedia_dev.man.movie*.

When release notes accompany a product, the product contains an additional *.man* image that contains the release notes—*dmedia_dev.man.relnotes*, in this case. Some reference page subsystems contain reference pages for more than one software subsystem. When you receive your workstation and when you install a software option for the first time, be sure to check the status of reference page subsystems to verify that the reference pages that you want are installed.

The commands that follow are useful for checking and installing reference page subsystems.

- Check to see which reference page subsystems are installed:

```
# showprods '*.man'
```

- List all reference page subsystems available for installation:

```
Inst> list '*.man'
```

- Select all reference page subsystems, not release notes, for installation:

```
Inst> install '*.man'
```

```
Inst> keep '*.relnotes'
```

Reinstalling an Older Software Release

Inst is designed to install software products that are new to your workstation or newer than the currently installed version. When you want to install a version of *oe* that is older than the currently installed version, the installation procedure must be modified. If you do not follow this modified procedure (for example, you use *set neweroverride on* instead), you may be left with configuration files that do not match the rest of the software. This can inhibit system reboots and make it difficult to diagnose compatibility problems.

The steps below describe the procedure for installing an older IRIX release after you have installed the new release. You should perform a complete system backup before performing this procedure.

In the procedure, you will back up your new, modified system configuration files, remove your new system software, and then install an older version. The final step is to integrate any modifications you made to your new configuration files into the restored, older configuration files.

Note: This procedure begins from the miniroot.

Before beginning this procedure, locate the older software distribution you will be downgrading to. Be sure that you have the installation tools (*sa* file) that accompanied the older distribution.

1. Load the miniroot from your current distribution.

Begin by loading the miniroot from your *current* distribution. (Refer to “Starting a Miniroot Installation” in Chapter 3 for information on how to load the miniroot.)

2. Invoke an IRIX shell.

Use the *shroot* command to escape to a root shell, enter *sh* to start a Bourne shell, and then use *showfiles* to create a list of modified configuration files:

```
Inst> shroot
# sh
# showfiles -s -c -m > /usr/tmp/configlist
```

3. Edit the configuration files list (optional).

If you want to edit the list of configuration files to add additional files or remove unnecessary files, you can do so now by setting your *TERM* variable and using *vi(1)*:

```
# TERM=vt100; export TERM
# vi /usr/tmp/configlist
```

4. Create backup configuration files.

When you are satisfied with */usr/tmp/configlist*, create the backup configuration files and then quickly replace the password and hosts files (in case of disaster while performing the rest of the procedure):

```
# sh -c 'while read fname; do mv $fname $fname.bak; done' <
/usr/tmp/configlist
# cp /etc/hosts.bak /etc/hosts
# cp /etc/passwd.bak /etc/passwd
# exit
# exit
Inst>
```

Note: The *< /usr/tmp/configlist* portion that appears on the second line of this example should be included at the end of the first line in your entry.

5. Remove target software.

Remove all currently installed software. At the Inst prompt, enter the commands shown below. It is critical that you not give any other commands at the Inst prompt at this time.

```
Inst> set rulesoverride true
Inst> view targ
Inst> remove I
Inst> go
```

If you receive a device-busy error at this point and receive the Interrupt menu, select *continue*. When Inst is finished and you quit, you may be asked if you really want to quit. Enter *y* for yes.

```
Inst> quit
Please wait ...
Ready to restart the system. Restart? { y(es), (n)o, (sh)ell, (h)elp
} sh
```

6. Enter the *sh* command to get a shell prompt.

7. Remove installation history files.

After escaping to the shell, remove all files that make up what is known as your installation history and then restart the system:

```
# rm -rf /root/var/inst
# exit
Ready to restart the system. Restart? { y(es), (n)o, (sh)ell, (h)elp
} y
```

8. Load the miniroot from the *older* distribution.

Use the older software distribution that you located at the beginning of this procedure to load the miniroot and invoke Inst; be sure that the installation tools and the software version match.

9. Install the older software version.

Remember that Inst no longer has valid software installation information about this system, so its initial selections are the default selections (see “Required and Default Subsystems” on page 7). Use Inst to specifically select for installation any additional software you wish to install.

10. Reboot your system.

11. Merge configuration files.

In IRIX, merge *config.bak* files (listed in */usr/tmp/configlist*) with the restored configuration files.

If you created a backup copy of customized configuration files (described in “Backing Up the Target Systems” on page 22) when you were running this release earlier, you should be able to restore the configuration files from that backup and use them. Add any new configuration information that has been added since they were last used.

If you have no backup of the current system files, compare the *.bak* and installed versions of the files in */usr/tmp/configlist*, then make the required changes using a process similar to the one for merging configuration files described in “Merging Configuration Files” on page 61. Be careful not to introduce new features from the later release.

If you modified any configuration files, you should restart the system now to boot from the desired configuration.

Avoiding Compatibility Problems

Although some compatibility information is contained in the product descriptions that Inst reads when it accesses a software distribution, Inst allows you to install incompatible software in many situations. The list below contains guidelines for avoiding compatibility problems.

- Inst tells you about incompatible subsystems when you try to install them. To learn about these incompatibilities earlier, check the release notes for the products that you plan to install.
- All subsystems of a product should have the same release number. When in doubt, give the *showprods* command with the *-n* argument:

```
% showprods -n names
```

The column that normally contains the installation date contains a number instead. Make sure that all subsystems in each product have the same number. If there are subsystems from different releases in the same product, compatibility problems might result.

- When you receive a software update, install the new versions of all the products you receive, not just some of them.
- If any subsystems that you want to install require a miniroot installation, perform all installations during a single miniroot installation session. Performing a single installation prevents incompatibilities that can occur when you switch from one installation to another.

Removing Software

You can remove software from a target system using the *versions remove* command (see the versions reference page). When you use this command, *inst* is invoked to perform the removal without user intervention. The subsystems, images, or products to be removed are specified as arguments to the command. For example, this command removes the entire 4DDN product and all release notes from the target:

```
# versions remove 4DDN *.man.relnotes
```

To remove all installed software from the target system, you must perform a miniroot installation; you cannot remove all software during a live install, since the miniroot is needed to supply the functions that sustain the system until critical software is replaced. Because removing all software is not desirable under most circumstances, *Inst* behavior is overridden to perform this procedure.

Follow these instructions to remove all installed software.

1. Override *Inst* rules:

```
Inst> set rulesoverride on
```

2. Select installed subsystems for removal:

```
Inst> remove I
```

3. Start the removal:

```
Inst> go
```

When you remove software, any configuration files that have been modified are not removed. This preserves any local information that might have been added. To list configuration files, enter the *showfiles -cCH* command from an IRIX shell.

Tailoring Installations

This chapter explains how to tailor installations using initialization scripts and the *set* command. Initialization scripts provide a programmatic interface to user-written scripts that execute when Inst or Software Manager receives a distribution specification. The *set* command assigns values to arguments known as *preferences*, which control aspects of the Inst and Software Manager interface. A table of available *set* preferences is included at the end of this chapter (a complete list is also available from Inst online help).

This chapter contains the following sections:

- “Using an Initialization Script” on page 112
- “Using README Files in a Distribution” on page 115
- “Setting Up an HTTP Installation Server” on page 116
- “Controlling Menu Displays” on page 120
- “Controlling Disk Space Checks and Reports” on page 121
- “Controlling the Display of Filenames” on page 123
- “Displaying the Current Distribution Source” on page 124
- “Turning Off Initial Selection of Subsystems” on page 125
- “Changing the Network Time-out Value” on page 125
- “A Closer Look at Preferences” on page 126

Note: The procedures in this chapter assume that *set* commands are entered from the Admin menu; however, except where noted, *set* commands can also be entered from the Main menu and View Commands menu.

Using an Initialization Script

The initialization script *inst.init* delays an installation session until the appropriate session parameters are set by the script. The primary function of *inst.init* is to redirect a distribution specification to a different source or to modify the default selections in a distribution. However, because an *inst.init* script can also include the use of selections files and *set* (preferences) commands, you can customize the script to meet a variety of installation requirements. The activation of *inst.init* is specified by the *startup_script* variable (see Table 8-1 for a description of this variable).

By default, if an *inst.init* script is located in a distribution directory, users are notified of the pending execution of the script whenever they specify the distribution directory during an installation session. Unless the user specifies otherwise, *inst.init* executes after the notification. If the specified distribution directory is a remote directory, *inst.init* is copied to the target system to execute (any scripts that *inst.init* calls are not copied to the target system, however).

For example, assume that a user on *babyface* enters one of these commands to install from server *easyboy*:

```
# inst -f easyboy:/d1/IRIX_6.3/dist
Inst> from easyboy:/d1/IRIX_6.3/dist
```

If *easyboy*'s */d1/IRIX_6.3/dist* contains a copy of *inst.init*, either of the previous commands copies and executes *inst.init* on *babyface* before the Inst Main menu is displayed. Similarly, *inst.init* is executed if it is located in the distribution directory when a Software Manager user uses the *Lookup* button.

Syntax of the *inst.init* Script

The *inst.init* script has this syntax:

```
inst.init [toolname] [pathname] [file descriptor]
```

Arguments have these meanings:

<i>toolname</i>	has the value <i>inst</i> or <i>swmgr</i>
<i>pathname</i>	is the specification for the directory that contains <i>inst.init</i>
<i>file descriptor</i>	is the file descriptor to which commands can be written

The *inst.init* script can read from *stdin* and write to *stdout* (file descriptor 1) using the controlling *tty* in Inst sessions; use *xconfirm*(1) for user input and output in Software Manager sessions. The script can also write to *stderr* (file descriptor 2). Messages to *stderr* are posted directly to the user by means of the *inst* pager during an Inst session or in the log pane during a Software Manager session. Messages to *stderr* are also copied to */var/inst/INSTLOG*, the installation log.

Note: Avoid using *stderr* to write prompts for an interactive Inst session. The *inst* pager might interfere with the session by writing unwanted lines to the screen.

Writing an *inst.init* Script

You can write *inst.init* as a shell script, a tcl script, or a binary program. The output of *inst.init* is a set of commands that Inst or Software Manager executes. These commands are supported as output from *inst.init*.

from [*distribution*]

specifies the location of the distribution. The *from* command is always executed first.

admin load [*selections file*]

specifies a selections file to be used for the installation (see “The save and load Commands” in Chapter 5 and the *-F* option of *inst*(1M)). You may use multiple selections files; in this case, all commands are buffered and executed in the order that they are read.

Note: Any *from* commands in selections files are ignored.

set [*preference*] [*value*]

sets preferences for Inst or Software Manager operation. Preferences are always set after the *from* command is executed.

install [*software.subsystem*]...

marks for installation the software products or product components that are specified as arguments.

keep [*software.subsystem*]...

maintains the current status of software products or product components that are specified as arguments.

remove [*software.subsystem*]...

marks software products or product components for removal.

Additional *inst.init* Script Rules

Keep these points in mind when creating an *inst.init* script:

- The *from* command executes first—the last *from* command returned by the script sets the distribution. All other commands are executed in the order in which the script sends them to *inst* for execution.
- All *from* commands in selections files are ignored.
- Lines written to the command descriptor must not begin with a space or tab character.
- Abbreviations and wildcards are supported in *install*, *keep*, and *remove* commands.
- Abbreviations of *from*, *set*, and *admin load* commands are not supported.

Sample *inst.init* Scripts

This sample *inst.init* script resets the distribution directory and notifies users of the change. Then the script retains all currently installed software in its current state, except InPerson; it installs all Inperson subsystems from the distribution except the reference pages.

Example 8-1 Sample *inst.init* Script

```
#!/sbin/sh
OLDPATH= $2
NEWPATH= yankee:/usr/dist
echo "$OLDPATH is unavailable"
echo "installing from $NEWPATH"
echo "from $NEWPATH" >&$3
echo "k *" >&$3
echo "i InPerson" >&$3
echo "k InPerson.man" >&$3
exit 0
```


This sample script determines the release and sets the distribution directory accordingly:

Example 8-2 Sample inst.init Script

```
#!/sbin/sh
#
# Run uname to determine the release and set the distribution directory
REL=`uname -r`
case $REL in
6.2*) echo "from $2/dist/6.2" >&$3 ;;
6.3*) echo "from $2/dist/6.3" >&$3 ;;
esac
```

Using README Files in a Distribution

When a *README* file is located in a distribution directory, the content of the file is displayed whenever users specify the distribution directory during an installation session. README contents might be special instructions about the software in this distribution, cautions, or other information that users might need before starting an installation session. If you plan to create a distribution, you can include three types of README files to provide information or instructions to users:

- *swmgr.README.html*
- *swmgr.README*
- *inst.README*

When a user starts an installation session from Software Manager and the distribution contains a *swmgr.README.html* file, a web browser displays the *swmgr.README.html* file. If the distribution contains a *swmgr.README* file but no *swmgr.README.html* file is present, or if the Web browser is not installed to view *swmgr.README.html*, the *swmgr.README* file is displayed at the start of the session. If the distribution contains neither a *swmgr.README.html* nor a *swmgr.README* file but it does contain an *inst.README* file, the *inst.README* file is displayed at the start of the session.

Users who start an installation session with Inst can be shown only an *inst.README file*. No other type of README file can be displayed at the start of Inst sessions.

Using a Related Distribution File

A related distributions file contains the names of additional distributions that are to be made available during an installation session. When a file named *.related_dists* is located in a directory, the software in the directory is considered the default distribution; the distributions in *.related_dists* are considered additional distributions that users can access using the *open* command. The *.related_dists* file can specify distribution directories on the network or additional CDs. For example, this file specifies two related distributions:

```
sparkplug:/dist/IRIX_6.3/options  
CDROM/dist
```

Setting Up an HTTP Installation Server

You can configure a server so that users are able to install software from it using a Web browser. Such installations use HyperText Transfer Protocol (HTTP) to establish the client-server connection and transfer the required files from the server to the client. HTTP installations are suitable for installing one or two products over the Web; they are not recommended for large installations, since HTTP connections do not provide the robustness of the TCP/IP connections normally used to install software across a network.

IRIX installation utilities support two methods for installing software over the Web:

- A selections file that references a *tardist* file, a Multipurpose Internet Mail Extension (MIME) file type that Web browsers recognize. With a *tardist* file, Software Manager performs a local installation, freeing the browser for other uses. For this reason, it is the preferred Web installation method.
- A Web distribution directory that contains product files and an index file listing the software that can be installed from the directory. Entries in the index file have an HyperText Markup Language (HTML) format. Web distribution directories are best suited to small product installations, because the browser is unavailable for other uses while the installation is in progress.

Using a Tardist Installation File

When you use a tardist file for Web installations, a selections file on the server contains a special entry that references the tardist file. The tardist file is created with the *tar* utility and contains all the distribution software required to install one or more software products. With this installation method, the URL on the user's Web page references the selections file containing the tardist file URL. (For more information on selections files, see "The save and load Commands" on page 77 and in the *inst(1M)* reference page.)

In tardist file installations, only the selections file is transferred over the HTTP connection. When the Web browser receives a selections file, it passes the file to Software Manager, using information in the MIME-type table. Software Manager opens a new HTTP connection to transfer the tardist file; then it unpacks the file onto a local temporary directory and installs the software from the local source. Because Software Manager uses a new HTTP connection, users are able to continue browsing while the installation is in progress.

Example 8-3 illustrates a selections file that references a tardist file on a Web server named *piston*.

Example 8-3 HTTP Selections File for a Tardist File Installation

```
from http://piston/distributions/dev_options.tardist
```

Note: The selections file can also include standard selections file entries, although none are shown in Example 8-3.

To set up the server for tardist file installations, follow these steps:

1. Create the tardist file and copy it to the server.

For example, these commands create a tardist file (*viewkit.tardist*) from the distribution source (directory */d1/distributions/options/ViewKit*) and copy the file to a Web server directory (*/var/www/htdocs/*):

```
# cd /d1/distributions/options/ViewKit
# tar cf viewkit.tardist *
# cp viewkit.tardist /var/www/distdir/
```

2. Create the selections file and copy it to the server.

Use your favorite editor to create a selections file like the one shown in Example 8-3. Remember that selections file must end with the *.inst* suffix. This example illustrates a selections file for the *tardist* file created in step 2:

```
# vi web_source.inst  
from http://piston/distdir/viewkit.tardist
```

This command copies the selections file to a public directory where it can be accessed from a Web browser:

```
# cp web_source.inst /var/www/htdocs
```

3. Write a Web page that references the selections file.

Create a Web page on the server from which users can install the software specified in the *tardist* file. The URL for the connection must reference the selections file (*/var/www/htdocs/web_source.inst* in the example from step 2).

Using an HTML Distribution Directory

An HTML distribution directory contains the software components needed to support one or more products and an index file that lists, in HTML format, the installable software. When installations are performed by this method, each entry in the index file is transferred to the client over the original HTTP connection. User-specified URLs to the directory must resolve to the server's index file.

The first line of an HTML installation file is a special entry that identifies the HTML format to Software Manager, ensuring that the file contents will be interpreted correctly. Subsequent entries specify the software components to be installed in HTML format, including the product description (the *spec* file) and the installation database (the *idb* file) for each product. (See Figure 1-1 on page 6 and "Anatomy of a Software Product" on page 4 for information on product components.) The directory that contains the HTML installation file must also contain the product components that it references.

Example 8-4 illustrates an HTML installation file that installs the ViewKit product. The first entry in this file is identical for all HTML installation files.

Example 8-4 HTML Installation File

```
@DISTRIBUTION.html Format# 02
<DL>
<DD>ViewKit
<DD>ViewKit.idb
<DD>ViewKit_eoe.man.relnotes
<DD>ViewKit_eoe.sw32.base
<DL>
```

To set up the server for tardist file installations, follow these steps:

1. Verify that the server is configured to return an index file.

The Web server must return a file to Software Manager. For example, if the URL is *http://piston/installs/*, the server must return *http://piston/installs/index.html*. Check server configuration; if it does not specify an index file, reconfigure it to do so.

2. Create the index file that contains the HTML installation entries.

Use your favorite editor to create this file. The first line of your file should be identical to the one in Example 8-4 on page 119: type a single space before and after `Format#`. Remember to include `spec` and `idb` file entries in the HTML file.

3. Copy the HTML installation file to the index file on the server.

In this example, the installation file `viewkit_dist.html` is copied to server's index file in the `/var/www/htdocs` directory:

```
cp index.html /var/www/htdocs/
```

Checking the Current Preference Settings

To see your current preferences settings, enter the `set` command with no arguments:

```
Inst> set
```

The output contains a line showing the name, the current setting, and a brief description of each preference.

For help on a particular preference, specify the preference in your entry:

```
Inst> set preference
```

The output contains a paragraph description of the preference, its name, type, and current value.

Controlling Menu Displays

By default, Inst displays the Main menu when it is invoked; menus show only a subset of the commands that can be entered at the `Inst>` and `Admin>` prompts. The following sections tell you how to change this default behavior.

Turning Off Menu Displays

To suppress the menu display when Inst is invoked, enter this command:

```
Admin> set menus off
```

When *menus* is set off, only the `Inst>` prompt is displayed when Inst is invoked. To return to the default behavior, enter this command:

```
Admin> set menus on
```

The value of *menus* is saved in the installation history database; its value is maintained for subsequent Inst sessions.

Displaying Hidden Commands

At each menu, several commands are available but not normally displayed. On the Inst Main menu, the hidden commands are *set*, *sh*, and *shroot*. On the Administrative Commands menu, the hidden commands are *help* and *quit*. On the View Commands menu, the hidden commands are *set*, *help*, and *quit*. The Interrupt menu contains no hidden commands; the *retry* command appears when Inst displays the Interrupt menu on errors, but *retry* is not available when you display the Interrupt menu manually.

Use this entry to set hidden commands on:

```
Inst> set fullmenu on
```

The value of *fullmenu* is saved in the installation history database; its value is maintained for subsequent Inst sessions.

Redisplaying the Menu

Because Inst does not redisplay the current menu after each command entry, you must request that the menu be redisplayed when you need it. The current menu is displayed if you press Enter (when the *menus* preference is set on) or if you enter a question mark (?). (The question mark is considered a pseudocommand because it never appears on menus.)

Controlling Disk Space Checks and Reports

Disk space checking enables Inst to make accurate estimates of whether disk space on a system is sufficient for the software selected for installation. The estimates help you determine which software to install or remove if disk space is limited.

Space calculations determine the amount of free disk space in all mounted filesystems, the disk space required for each subsystem, the temporary requirement for disk space during the installation process, and the net change in disk space if a subsystem is installed or removed. If a detailed space check is requested, Inst also checks the disk for any deleted files, and for any local modifications to the installed software.

You see this message when Inst performs space checking:

```
Calculating sizes:
```

The space checks performed during *list* and *step* processing calculate the net amount of space that your selections will require after installation. The space checks performed during *go* processing calculate the net space requirement and the additional space that will be required during the installation process.

Estimated and Detailed Space Checks

Inst performs two types of disk space calculations: *estimated space checks*, which are based on the information in the Inst history file, and *detailed space checks*, in which Inst accesses file size information on the disk to perform its calculations. By default, estimated space checks are performed when you use the *from* command to change the distribution source and when you use the *list* or *step* command to make selections. Detailed space checking is always performed after you enter the *go* command to initiate installation. You can change these defaults using *set* command preferences.

Note: You can disable all space checking with the *space_check* preference, but disabling space checking is not advised (see Inst help for details).

Changing Detailed Space Checking

There may be times when you need very accurate disk space calculations throughout an installation session (when a target has limited disk space, for example). In this case, you can use the *detailSPACEcheck* preference of the *set* command to force detailed space checking to be performed during *list* and *step* processing as well as during *go* processing. To force additional detailed space checking, enter this *set* command before you enter the *list* or *step* command in the session:

```
Admin> set detailSPACEcheck on
```

Inst saves the value of *detailSPACEcheck* in the installation history, so it is not necessary to reset this preference if you wish to maintain this level of checking in subsequent sessions. However, increasing the number of detailed space checks also increases the amount of time required for the interactive portion of an installation session (the amount of time required for *go* processing remains the same, regardless of the *detailSPACEcheck* setting).

To return to the default space-checking behavior, enter this command:

```
Admin> set detailSPACEcheck off
```

Delaying Disk Space Checking

The *delaySPACEcheck* preference of the *set* command allows you to postpone disk space checking until after you enter the *go* command. Delaying space checking reduces the amount of time needed for the interactive portion of an installation session. However, when you delay space checking, disk space information is not displayed in the output of *list* and *step* commands—subsystem sizes are listed as zero.

To postpone disk space checking until you enter the *go* command, enter this command before entering the first *list* or *step* command:

```
Admin> set delaySPACEcheck on
```

Inst saves the value of *delaySPACEcheck* in the installation history database, so it is not necessary to reset this preference if you wish to delay space checking in subsequent installations.

To return to the default space-checking behavior, enter this command:

```
Admin> set delayspacecheck off
```

Minimizing Disk Space-Checking Time

When you want to install only one product and the distribution source contains many products, you can reduce the amount of space-checking time by including the product name when specifying the distribution source. When you specify a product in the distribution source, checking is limited to the disk space requirements of the individual product rather than the entire distribution.

These entries illustrate how to specify an individual product as the distribution source:

```
# inst -f source/product
Inst> from source/product
Inst> from $dist/product
```

The *from* command must be entered before a *list* or *step* command that triggers disk space checking (the *source*, *product*, and *\$dist* arguments are described in “Alternative Ways to Specify a Distribution” on page 83).

Controlling the Display of Filenames

By default, Inst does not report the names of files as it installs and removes them. The *set* preferences described below allow you to modify this behavior and control the way that filenames are displayed.

To display filenames as they are installed, enter this command before you enter *go*:

```
Admin> set show_files on
```

To suppress filenames as they are installed, enter this command at any prompt:

```
Admin> set show_files off
```

The value of *show_files* is saved in your installation history database, so it has the same value from one installation session to the next until you change it.

By default, when filenames are displayed during installation and removal, each filename replaces the one that was displayed before it. To change the display so that each name is displayed on a new line, enter this command:

```
Admin> set overprint off
```

To return to the default behavior, enter this command:

```
Inst> set overprint on
```

The value of *overprint* is saved in the installation history database, so it is not necessary to reset it in subsequent installation sessions.

Displaying the Current Distribution Source

The current distribution source is displayed in a message above the Main menu when Inst is invoked. The message has one of these forms:

```
Default distribution to install from: source  
Default distribution to install from: source/product
```

Depending on your situation, *source* includes one or more of these items: a user ID, a remote target name, and a directory or tape device. If you specified a product name when you invoked Inst (*inst -f*), *product* is the name of the software product that you specified (for live installations only). The complete syntax of *source* and *source/product* is entered in “Step 2: Specifying the Source” on page 41 and “Alternative Ways to Specify a Distribution” on page 83.

Each time you enter the *from* command, the current distribution source changes. You can display the current distribution source by entering this command from any menu:

```
Inst> set dist  
Preference          Type          Value  
...  
dist                 String        last_source  
Inst>
```

The value of *dist* is *last_source*, which is a character string (*string*). The value of *last_source* is always the same as *source* (the current distribution), with one exception: After Inst is invoked, but before product descriptors are read for the first time, the value of *dist* is the distribution source that was specified in the previous installation session. If you use a distribution source of the form *source/product*, *product* is not included in the value of *dist*.

Turning Off Initial Selection of Subsystems

Each time you invoke `Inst` or enter a `from` command, `Inst` ignores any installation or removal selections that were not performed; it uses its own algorithms to select subsystems for installation. You can override this selection of subsystems, but any pending installation or removal selection is still erased.

To prevent `Inst` from making selections after you enter the `from` command, enter this command:

```
Inst> set autoselect off
```

Turning `autoselect` off means that the only *i* characters in the first column of `list` output are selections that you made. To turn automatic selection back on, use this command:

```
Inst> set autoselect on
```

The value of `autoselect` is saved in the installation history database; if you set it to off, it remains off in future `Inst` sessions until you change its value.

Changing the Network Time-out Value

`Inst` allows you to specify a network time-out value. This value specifies the length of time that `Inst` waits for responses from a remote system before giving up.

Use the `set` command to display and set the time-out value:

```
Admin> set
Preference Value Description
...
timeout 180 Network timeout, in seconds
...
Inst> set timeout 120
```

A minimum value of 120 seconds is recommended. The value of `timeout` is saved in the installation history database; when you find a value of `timeout` that works for your site, you can leave it permanently set (unless network conditions change and time-out errors begin to occur).

A Closer Look at Preferences

Preferences allow you to tailor the installation utility so that it operates suitably for a single user or user group. In most cases, preference settings apply to both Inst and Software Manager (exceptions are *tty* preferences, which apply exclusively to Inst, and *gui* preferences, which apply exclusively to Software Manager).

Preferences are classified on the basis of their duration. *Permanent* preferences retain their settings until they are reset; *transient* preferences revert to a default setting at the start of a new installation session. A third group of preferences have the potential to cause installation errors when set arbitrarily; these preferences are classified as *expert*.

Most preferences are assigned simple Boolean values; that is, you use the *set* command to turn them on or off. You can also substitute *true* (for *on*) and *false* (for *off*) when setting these preferences. These examples illustrate equivalent command pairs:

```
Admin> set beep on
Admin> set beep true

Admin> set short_names off
Admin> set short_name false
```

The Preferences File

In the factory-shipped configuration of *inst*, preference settings are stored in the */var/inst/resources* file. However, whenever the original setting for a permanent preference is changed, the new setting is recorded in the *.swmgrrc* file, which is automatically created at the first change. The automatically generated copy is stored in the root directory, *\$rbase*, for the new software installation at *\$rbase/var/inst/.swmgrrc*.

Each line of the *.swmgrrc* file contains the name of a preference and its assigned values. Example 8-5 illustrates the format of *.swmgrrc* entries. It illustrates settings that are expressed as Boolean, single choice, multivalued, and integer values.

Example 8-5 Sample *.swmgrrc* File

```
auto_inst_new: true
display_size: blocks
inst_visible_resources: tty,transient,permanent,beginner,expert
network_retries: 4
```

Creating Additional Preferences Files

You can create and store additional copies of *.swmgrrc* to tailor preferences for a particular system or for individual users. When multiple copies of *.swmgrrc* exist, the settings that they contain are added to the settings in *\$rbase/var/inst/.swmgrrc*. If conflicting settings are recorded for a particular preference, the most user-specific setting takes precedence, as explained below.

Individual users can create additional *.swmgrrc* files in these locations, in order of highest to lowest precedence:

\$HOME/.swmgrrc

Contains preference settings for all installation sessions that the user initiates (except miniroot installations) and resides in the user's home directory on the target system. Takes precedence over settings in *\$rbase/var/inst/.swmgrrc*, */var/inst/.swmgrrc*, and *\$rbase/\$HOME/.swmgrrc*.

\$rbase/\$HOME/.swmgrrc

Contains an individual user's preference settings for miniroot installations and resides in a special home directory for the installation (a subdirectory of the root directory for the newly installed software, such as *\$rbase/var/people/\$HOME*, for example). Takes precedence over settings in *\$rbase/var/inst/.swmgrrc* and */var/inst/.swmgrrc*.

/var/inst/.swmgrrc

Contains preference settings that are used for all installations performed on the target by any user. Whereas the effects of the *\$rbase/var/inst/.swmgrrc* file are restricted to the root directory for the new software (*\$rbase*), */var/inst/.swmgrrc* affects installations to any filesystem on the target.

When *Inst* or *Software Manager* is invoked, existing *.swmgrrc* files are read in this order:

1. */var/inst/.swmgrrc*
2. *\$rbase/var/inst/.swmgrrc*
3. *\$rbase/\$HOME/.swmgrrc*
4. *\$HOME/.swmgrrc*

When preference settings are added to *.swmgrrc*, the new values are recorded in the file that was opened last. As a result, preference settings are recorded in the most localized version of the *.swgrrc* file that is open during the installation session.

The Preferences List

The tables in this section describe the preferences that you can specify with the *set* command or from the Customize Installation menus in Software Manager to control the installation process. They list the name of the preference, describe the function that the preference performs, and list the default setting for the preference. Expert preferences are identified in the first column. The fourth column, *Value Saved?* indicates whether the preference remains in effect for subsequent installation sessions (*yes*) or reverts to a default value at the start of a new session (*no*).

Table 8-1 contains preferences that affect both Inst and Software Manager.

Note: Some preferences in this table contain advisories in the Function column. Be sure to read these advisories carefully before resetting the preference to which they pertain.

Table 8-1 Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
abort_on_error (Expert)	Controls whether an automatic installation is aborted if an error occurs. If <i>true</i> (default), an error cancels the installation; if <i>false</i> , the installation continues after the error occurs. Regardless of setting, errors are reported in <i>/var/inst/INSTLOG</i> .	true	yes
always_confirm_quit	Controls whether the user is always asked to confirm a <i>quit</i> command. If <i>true</i> , user must confirm a quit; if <i>false</i> (default), no user confirmation is needed after <i>quit</i> .	false	yes
autoconfig_overhead (Expert)	Controls the amount of disk space reserved for building the IRIX kernel (also see the <i>kernel_size_32</i> and <i>kernel_size_64</i> preferences).	179.2 MB	no
autoselect	Controls whether <i>inst</i> does an automatic selection of subsystems during initialization and when a new distribution source is specified. If <i>true</i> (default), <i>inst</i> automatically selects subsystems at these events in the session; if <i>false</i> , no subsystems are automatically selected.	true	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
beep (Expert)	Controls whether an audible signal is given after <i>go</i> processing is complete. If <i>true</i> (default), audible signal rings when <i>go</i> processing finishes; if <i>false</i> , no signal is given.	true	yes
broadcast (Expert)	Allows you to specify a port on a remote host where UDP messages will be posted during a miniroot installation so that the installation can be monitored remotely. Specify the IP address of the host and a port number (if different from the default) where the messages will be posted. The format of the specification is <i>host[:port]</i>	N/A	no
checkpoint_restart (Expert)	Set during a session recovery when the user selects the action to take after an abnormal exit. If <i>true</i> , the user chose to retry the interrupted installation using the selections made during the session; if <i>false</i> (default), the user chose either to return to the previous session without attempting to install selected software or ignore the previous session completely.	false	no
checkpoint_selections (Expert)	Controls whether installation and removal selections are saved in the checkpoint file so they can be used in a session recovery. If <i>true</i> , user selections are saved in the file as soon as they are made; if <i>false</i> (default), pending selections are saved when the user enters the <i>go</i> command.	false	yes
confirm_quit	Controls the confirmation prompt for <i>quit</i> commands when selections are pending. If <i>true</i> (default), users are notified if install or removal selections are pending when they enter <i>quit</i> ; if <i>false</i> , users are allowed to quit the session without a reminder of pending selections.	true	yes
confirm_nfs_installs	Controls whether user is prompted to confirm installation on NFS mounted filesystems. If <i>on</i> (default), user is prompted for confirmation; if <i>off</i> , files are installed in the NFS filesystem if the user has proper permissions.	on	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
default_config (Expert)	Controls handling of configuration files. If <i>true</i> , all configuration files in the distribution are installed; if the target version was modified, it is saved in a file with a <i>.O</i> extension. If <i>false</i> (default) and the target version was modified, install the distribution version of the configuration files, appending a <i>.N</i> extension; if the target version was not modified, overwrite it with the new file.	false	yes
default_sharedirs (Expert)	For diskless systems, contains the list of default shared directories used during share-tree and client-tree installations.	/usr	no
delayspacecheck	Controls the timing of disk space calculations. If <i>on</i> , disk space calculations are deferred until the <i>go</i> or <i>space</i> commands are entered and no space information is provided when <i>list</i> or <i>step</i> commands are entered; if <i>off</i> (default), space calculations are performed at the first <i>list</i> , <i>step</i> , or <i>space</i> command, unless disk space checking was done previously.	off	yes
detailspacecheck	Controls the source of the data used in space checking. If <i>on</i> , space requirements for files are determined by checking their sizes on the disk; if <i>off</i> (default), file sizes are assumed to be the value that is recorded in the installation history database. Advisory: Set <i>on</i> for slower, more accurate calculations.	off	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
delay_conflicts	Controls whether messages for delayable conflicts (those caused by incompatible software selections), normally posted before installation is performed, are delayed until Inst or Software Manager exit. Valid settings are <i>ask</i> , <i>on</i> , and <i>off</i> . If <i>ask</i> (default), the user is prompted after the first delayable conflict to specify whether subsequent conflict messages should be delayed; the specification remains in effect for the remainder of the session unless the user changes it. If <i>on</i> , delayable conflicts are always postponed until Inst or Software Manager exits. If <i>off</i> , conflicts are presented before software installation.	ask	no
delay_idb_read	Delays reading of the installation database (idb) files until after the <i>go</i> or <i>Start</i> (for Software Manager) command is issued, increasing the speed of reading large distributions. If <i>default</i> , reading is delayed unless the distribution medium is a local CD. If <i>on</i> , reading is always delayed until after <i>go</i> or <i>Start</i> , regardless of the distribution medium. If <i>off</i> , idb files are always read as soon as a command is issued that requires idb file information. Advisory: Set to on if users are installing from a single CD or if users are comfortable with multiple CD changes during the session.	default	no
delay_exitops	Controls when exit operations are performed. If <i>true</i> , exit operations are performed after users quit the installation session; if <i>false</i> (the default), exit operations are performed after all software is installed but before control is returned to the user.	false	yes
disable_keepfile	Suppresses processing of <i>\$rbase/var/inst/.keepfile</i> , which can be created to alter the initial software selections that <i>inst</i> makes at the beginning of an installation session. If true, <i>.keepfile</i> is disabled; if false (the default), <i>.keepfile</i> remains in effect.	false	no

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
dist	Specifies the value of the current distribution source. For live installations, the default value is the previously specified source. For miniroot installations, <i>dist</i> must be specified. Product names are not saved as part of the value.	<i>source</i>	yes
exclusions (Expert)	Specifies a list of files on which no action is taken during <i>go</i> or <i>Start</i> (for Software Manager) processing (also see <i>restrictions</i> preference). If a directory is specified, all files that it contains are also excluded from <i>go</i> and <i>Start</i> processing.	N/A	no
hide_image_products (Expert)	Controls whether the image components of each product are displayed when the product list is displayed. If <i>true</i> (default), listings of products do not include their component images. If <i>false</i> , the image components of a product are displayed in product listings.	true	yes
inst_initial_level (Expert)	Controls the initial setting of the <i>view</i> command. If <i>subsystem</i> (default), the output of <i>list</i> displays the subsystems in each product that appears in the listing; if <i>image</i> , the output of <i>list</i> displays the images for each product that appears in the listing; if <i>product</i> , the output of <i>list</i> displays product names without product components.	<i>subsystem</i>	yes
install_identical_files (Expert)	Controls whether files are installed when the distribution version appears identical to the version installed on the target. If <i>true</i> (default), identical distribution files are installed. If <i>false</i> , distribution files are not installed when they appear identical.	true	yes
install_sites (Expert)	Contains the list of previous distribution sources (also see the <i>install_site_size</i> preference).	N/A	yes
install_sites_size (Expert)	Specifies the number of distribution sources that are maintained in the installation sources list (also see the <i>install_sites</i> preference).	10	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
kernel_size_32 (Expert)	Specifies the amount of disk space that is reserved for building a new kernel on 32-bit systems. See also the <i>autoconfig_overhead</i> and <i>kernel_size_64</i> preferences.	393.2 MB	no
kernel_size_64 (Expert)	Specifies the amount of disk space that is reserved for building a new kernel on 64-bit systems. See also the <i>autoconfig_overhead</i> and <i>kernel_size_32</i> preferences.	576.7 MB	no
live_install	Controls how operating system installations are performed when the session is started with the system in multi-user state. If <i>default</i> , the system is automatically shut down for operating system software installations; when the installer enters <i>go</i> or <i>Start</i> , users receive a warning that the system is shutting down. If <i>on</i> , installing operating system software is permitted in multi-user state. If <i>off</i> , the system always attempts to load the miniroot; if the distribution contains no standalone IRIX file (<i>sa</i>), the installation fails an error message is posted. Advisory: This preference is valid only for installations into the / directory.		
neweroverride	Controls whether older products can replace newer ones. If <i>on</i> , users can install older products in place of new ones; if <i>off</i> (default), older products cannot be installed when a more recent version exists on the target.	off	no
override_space_check (Expert)	Controls whether space checking is performed. If <i>true</i> , no space checking is done during an installation, allowing disk overflows. If <i>false</i> (default), space checking is performed. Advisory: Setting <i>true</i> can cause serious installation problems.	false	no

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
post_install_dialog (Expert)	Controls whether a completion notice is posted when an installation is completed. If <i>true</i> (default), a completion notice is posted. If <i>false</i> , the user sees a message only if a reboot is required.	true	yes
precious_files (Expert)	Specifies a list of files that will not be overwritten or removed by the installation.	N/A	no
promptforid (Expert)	Controls whether user is prompted to specify unknown user IDs (UIDs) and group IDs (GIDs). If <i>true</i> , user is prompted for a UID if one is not found in <i>/etc/passwd</i> and a GID if one is not found in <i>/etc/group</i> . If <i>false</i> (default), inst assigns a UID and GID based on the <i>startid</i> preference and writes them to <i>/usr/adm/SYSLOG</i> .	false	yes
restrictions (Expert)	Specifies a list of files that are the only ones to be installed or removed during <i>go</i> or <i>Start</i> (for Software Manager) processing (see also the <i>exclusions</i> preference). If a directory is specified, all files that it contains are included in the restricted list. No action is taken on any files that are not specified, regardless of their selection status when the installation is launched.	N/A	no
rulesoverride	Controls whether conflicts can be overwritten. If <i>on</i> , incompatible subsystems and subsystems without prerequisites may be installed on the target. If <i>off</i> (default), incompatibilities and prerequisites must be resolved before installation is done. Advisory: Use <i>on</i> setting with caution.	off	no
set_path (Expert)	Specifies the value to use when setting the <i>\$PATH</i> variable for exit operations. If <i>default</i> (default), set to a known path; if <i>environment</i> , set path to <i>\$PATH</i> environment variable or use the default path if <i>\$PATH</i> is not set; if <i>specific</i> , use the value in <i>set_path_specific</i> preference.	default	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
set_path_specific (Expert)	Specifies the path to use for exit operations. Value is a user-supplied string.	N/A	yes
shell	Specifies the program to use for the <i>sh</i> and <i>shroot</i> Administrative commands. Usually this value is either <i>/bin/sh</i> or <i>/bin/csh</i> .	<i>/bin/csh</i>	yes
short_names	Controls product names in listings. If <i>true</i> , only the shortname is displayed in listings; if <i>false</i> (default), product descriptions are displayed instead of their shortnames.	false	yes
show_absolute_sizes	Sets <i>list</i> and <i>step</i> displays to provide absolute subsystem sizes rather than relative sizes (a comparison between the size of the distribution subsystem and its corresponding subsystem on the target). If <i>on</i> , the absolute subsystem size is displayed; if <i>off</i> (default) the relative size is displayed. Advisory: Default unit of measure is kilobytes. See also <i>display_size</i> .	off	no
show_files (Expert)	Controls whether filenames are displayed during installation. If <i>true</i> , each component file in a product is displayed as it is installed; if <i>false</i> (default), the product name is displayed as it is installed.	false	yes
show_hidden_resources (Expert)	Controls whether user is shown expert preferences in addition to basic preferences. If set to <i>false</i> (default), user sees only basic preferences; if set to <i>true</i> , user sees expert and basic preferences.	false	yes
show_lint (Expert)	Determines whether inconsistencies in the distribution are displayed during reading (for debugging). If <i>true</i> , inconsistencies are displayed; if <i>false</i> (default), inconsistencies are not reported.	false	yes

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
space_check (Expert)	Controls whether space checking is performed. If <i>true</i> (default), space checking is done at various points in the installation (see the <i>delayspacecheck</i> preference). If <i>false</i> , no space checking is performed. Advisory: Setting <i>false</i> can cause serious installation problems.	true	no
startid	Sets the starting value of UIDs and GIDs when this information is unknown to <i>inst</i> and <i>promptforid</i> preference is set off (also see <i>promptforid</i> preference). The value of <i>startid</i> is assigned to the first unknown UID and GID; subsequent unknown UIDs and GIDs are incremented by one.	60000	yes
stream	Switches between the maintenance release stream and the feature release stream. This preference is initially set the first time that you make a stream selection.	N/A	yes
startup_script	Controls how Inst and Software Manager treat the initialization script file, <i>inst.init</i> . If set to <i>prompt</i> (default), the user is prompted before the script file is executed and the README file is displayed if one exists; for noninteractive installations in which the setting is <i>prompt</i> and the initialization file exists, Inst or Software Manager exits. If set to <i>ignore</i> , <i>inst.init</i> is always ignored. If set to <i>execute</i> , <i>inst.init</i> always executes without prompting the user.	prompt	no
swmgrrc_path (Expert)	Specifies the search path for the <i>.swmgrrc</i> file. The default path is /var/inst:\$rbase/var/inst:\$rbase\$HOME: \$HOME	(See Function column.)	no

Table 8-1 (continued) Preferences That Affect Inst and Software Manager

Preference Name	Function	Default Setting	Value Saved?
timeout	Specifies the amount of time, in seconds, that <i>inst</i> allows for a reply from a remote host before retrying the connection. Two retries are done after the initial attempt. Advisory: Set to a value that is suitable for your network.	120 seconds	yes
verbose (Expert)	Works with <i>verbosity</i> preference to control the number of information messages that appear during installation sessions (see also <i>verbosity</i>). If <i>off</i> (default), messages are set to verbosity level 2; if <i>on</i> , messages are set to verbosity level 4. If both <i>verbose</i> and <i>verbosity</i> settings are changed, the most recent setting takes precedence.	off	yes
verbosity (Expert)	Controls the number of informational messages that are displayed during an installation session (see also <i>verbose</i>). Valid settings are 0, silent; 2, verbose off; 4, verbose on; 6, all generated messages. When <i>verbose</i> and <i>verbosity</i> are changed, the most recent setting takes precedence.	2	yes
verify_checksum (Expert)	Controls whether files are verified by <i>checksum</i> as they are uncompressed or read from the archives. If <i>true</i> (default), <i>checksum</i> verification is performed. If <i>false</i> , no checksum verification is made.	true	yes

Table 8-2 contains preferences that affect Inst only.

Note: Some preferences in this table contain advisories in the Function column. Be sure to read these advisories carefully before resetting the preference to which they pertain.

Table 8-2 Preferences That Affect Inst Only

Preference Name	Function	Default Setting	Value Saved?
always_page_inst	Controls whether requested information, such as progress messages during <i>go</i> processing, are displayed in page segments or scrolled off the screen. If <i>true</i> , requested information is displayed in page segments; if <i>false</i> (default), information that is longer than a page scrolls off the screen.	false	no
clearprompt	Controls how throwaway prompts are displayed. If <i>true</i> (default), these prompts are cleared from a line by overwriting with blank spaces; if <i>false</i> , the original prompts remain on the line and a new line is generated to display a new prompt. Advisory: Set to <i>false</i> on terminals that do not perform line clearing.	true	yes
display_size	Controls the unit of measure for subsystem sizes. Valid settings are <i>kbytes</i> (default), <i>kilobytes</i> , <i>bytes</i> , or <i>512-byte blocks</i> .	kbytes	no
fullmenu	Controls the display of hidden commands (but not their availability) on the Main menu, Administrative Commands menu, and View Commands menu. If <i>on</i> , hidden commands appear on these menus; if <i>off</i> (default), hidden commands do not appear.	off	yes

Table 8-2 (continued) Preferences That Affect Inst Only

Preference Name	Function	Default Setting	Value Saved?
inst_terse_mode (Expert)	Restricts messages that are displayed on the screen. If <i>false</i> (default), user sees standard status messages, warnings, and errors. If <i>true</i> , most status messages are repressed and only warnings, errors, and output requested by the user are posted. <i>True</i> also forces off <i>page_output</i> , <i>show_diskspace</i> , <i>show_legend</i> , <i>show_percent_done</i> , and <i>show_files</i> and sets <i>verbosity</i> to 1. See also <i>verbose</i> preference, <i>admin save</i> and <i>admin load</i> commands, and the <i>inst</i> (1M) option <i>-F <selections file></i> .	false	yes
inst_visible_resources	Controls the list of preferences that are displayed when the <i>set</i> command is entered with no arguments (see also <i>swmgr_visible_resources</i>). This preference takes multiple values: <i>tty</i> (default) specifies that the displayed list should contain all preferences that apply to the Inst interface; <i>permanent</i> (default) specifies that the displayed list should contain all preferences whose value is maintained in subsequent sessions (until the value is reset); <i>transient</i> specifies that the displayed list should contain all preferences whose value reverts to a default at the end of a session; <i>beginner</i> (default) limits the displayed list to all preferences that are not designated <i>expert</i> ; <i>expert</i> specifies that the displayed list contain all preferences whose settings have potentially adverse effects on installation. Advisory: In your entry, separate values with a comma; no spaces are allowed.	<i>tty</i> permanent transient beginner	yes

Table 8-2 (continued) Preferences That Affect Inst Only

Preference Name	Function	Default Setting	Value Saved?
menus	Controls whether menus are displayed automatically. If <i>on</i> (default), Main menu is displayed at startup and all menus are displayed when they are invoked; if <i>off</i> , menus are not displayed at startup or invocation, but users can display them manually by entering ? at the menu prompt. Advisory: Set <i>off</i> for experienced Inst users.	on	yes
miniroot_graphics	Controls whether progress messages sent from the miniroot are displayed as ASCII text or graphics. If <i>default</i> , only miniroot installations launched from Software Manager use graphical displays; if <i>on</i> , all miniroot installations use graphical displays; if <i>off</i> , no miniroot installations use graphical displays.	default	no
overprint	Controls the display of the software listing during installation (or removal). If <i>on</i> (default), the same line is used to display the name of each software item as it is installed; the name is overwritten when the next item is installed. If <i>off</i> , a new line is used to display each name; the list scrolls down the screen.	on	yes
page_output	Controls tty device output. If <i>true</i> (default), output is managed by paging that is similar to <i>more</i> (see the <i>more(1M)</i> reference page); if <i>false</i> , output scrolls off the screen.	true	no
show_diskspace	Controls whether a disk space summary is displayed by the <i>list</i> , <i>step</i> , and <i>recalculate</i> commands. If <i>true</i> (default), the disk space summary is displayed. If <i>false</i> , disk space summaries are suppressed for <i>list</i> , <i>step</i> , and <i>recalculate</i> commands, but can be displayed with the <i>admin space</i> command. See also <i>inst_terse_mode</i> .	true	yes

Table 8-2 (continued) Preferences That Affect Inst Only

Preference Name	Function	Default Setting	Value Saved?
show_legend	Controls the legend for <i>list</i> displays. If <i>true</i> (default), the output of <i>list</i> contains an explanation of the mnemonic tags that appear in the listing; if <i>false</i> , the legend is not posted.	true	yes
show_percent_done (Expert)	Controls whether task completion messages are posted. If <i>true</i> (default), Inst posts periodic messages during various processing tasks to report the percentage of the task that is completed. If <i>false</i> , no task completion messages are posted.	true	no
wrapmode	Controls whether displayed lines will wrap or truncate at the screen width limit. If <i>wrap</i> (default), lines longer than the screen width wrap to the next line; if <i>truncate</i> , lines are truncated at the limit of the screen width.	wrap	yes

Table 8-3 contains preferences that affect Software Manager only.

Table 8-3 Preferences That Affect Software Manager Only

Preference Name	Function	Default Setting	Value Saved?
background (Expert)	Controls whether Software Manager is run as a background or foreground task when invoked from an IRIX shell. If <i>true</i> (default), Software Manager runs as a background task; if <i>false</i> , Software Manager runs as a foreground task.	true	yes
custom_startup_mode (Expert)	Controls how Software Manager is initially presented. If <i>off</i> (default), Software Manager always starts in Automatic Installation mode (selections are automatically made at initialization). If <i>distribution</i> , it starts in Customize Installation mode (the user makes selections) if a distribution was specified; if no distribution was specified, it starts in Automatic Installation mode. If <i>always</i> , Inst starts in Customize Installation mode if a distribution was specified; if no distribution was specified, it starts in Manage Installed Software mode (actions apply to target software only).	off	yes
never_resize_pane (Expert)	Controls whether pane is resized when switching the selection mode (see <i>custom_startup_mode</i> preference). If <i>true</i> (default), the pane size remains the same for all selection modes. If <i>false</i> , pane size is changed when the selection modes are changed.	true	yes
space_update_interval (Expert)	Specifies in seconds the frequency with which the Disk Space area is automatically updated; these updates are in addition to the updates that occur when product selections are change. Valid settings are integers; updates are disabled if set to <i>0</i> .	10	yes

Table 8-3 (continued) Preferences That Affect Software Manager Only

Preference Name	Function	Default Setting	Value Saved?
swmgr_initial_level (Expert)	Controls the level of the product hierarchy that is displayed when Software Manager is invoked. Valid settings are <i>product</i> (default) and <i>subsystem</i> .	product	yes
swmgr_visible_resources	Controls the set of preferences that are visible in the Preferences dialog (see also <i>inst_visible_resources</i>). These settings are valid: if <i>none</i> , no preferences are displayed in the dialog; if <i>permanent</i> (default), the dialog contains all preferences whose values are retained for subsequent installation sessions (until the value is changed); if <i>transient</i> , the dialog contains preferences whose settings revert to a default at the end of a session; if <i>beginner</i> (default), the dialog contains all preferences that are not designated <i>expert</i> ; if <i>expert</i> , the dialog contains preferences whose settings have potentially adverse effects on installation.	beginner permanent	yes
use_last_dist (Expert)	Controls the contents of the Available Software field. If <i>true</i> , the last distribution source that was used is automatically entered in this field at startup. If <i>false</i> (default), this field is empty at startup.	false	yes

PART TWO

RoboInst

Chapter 9, “Automating Installations With RoboInst”

Automating Installations With RoboInst

RoboInst is an extension to the miniroot that automates miniroot installations and performs related tasks, such as disk and filesystem configuration, as an integral part of the installation process. RoboInst also allows a “live-mode” installation as described in “Additional RoboInst Command-Line Options” on page 165. The tasks performed during the installation are user-defined. RoboInst can be launched locally on the target host (RoboInst client) or from a network server (RoboInst server) to install any number of clients located anywhere on an internet. It can also be launched by a scheduler such as *cron* or from a batch queue.

Note: A RoboInst license is purchased separately. Refer to Chapter 10, “Licensing,” for information on acquiring a license to use RoboInst.

This chapter describes RoboInst fundamentals, how to prepare for using RoboInst, and how to customize miniroot installations with RoboInst scripts. The chapter contains these sections:

- “How RoboInst Works” on page 147
- “Getting Ready for RoboInst” on page 151
- “Customizing Miniroot Installations” on page 154
- “Launching RoboInst” on page 162

How RoboInst Works

Functionally, a RoboInst configuration server can be seen as four servers and any number of clients. The servers are:

- RoboInst server—Runs *roboinst* command.
- software distribution server—Contains distribution(s) to be installed.
- configuration server—Contains RoboInst configuration file(s).
- boot server—Contains *sash* and miniroot files.

The installation is initiated from a RoboInst server, which may also be a boot server, distribution server, or client. The distribution to be installed is provided by a software distribution server (also called an installation server) which may or may not also be a boot server and/or configuration server. The clients are any network hosts that receive the installation.

The sequence of events in a RoboInst installation are specified by various files that reside in the *configuration directory*; the host containing the configuration directory is the configuration server. The configuration directory must contain a master miniroot configuration file, *mrconfig*, but it can also contain additional files and scripts that support the installation process. All files and scripts in the configuration directory, including *mrconfig*, are user-created; they specify actions to be taken before, during, or after the miniroot installation. See “Creating a Miniroot Configuration File” on page 154 for details.

Like other remote miniroot installations, RoboInst installations frequently rely on a boot server to load the miniroot from a remote source to the client system (see “About Miniroot Installations” on page 29). The boot server contains a *sash* and IRIX miniroot boot files for all system models to be installed. In a simple RoboInst configuration, the boot server and the configuration server reside on the same host, but they can reside on separate hosts if necessary.

Figure 9-1 illustrates a simple RoboInst configuration.

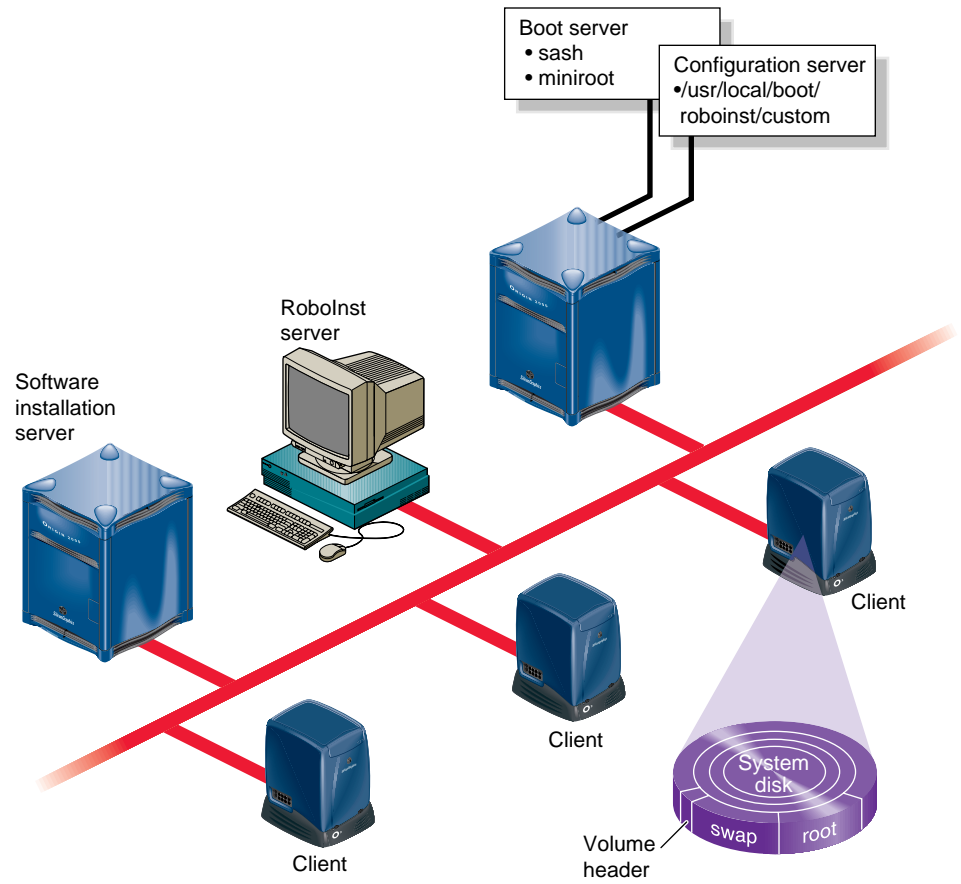


Figure 9-1 Simple RoboInst Configuration

Events in a RoboInst Installation

To perform an automatic miniroot installation on one or more clients, issue a *roboinst* command from the RoboInst server. This sequence of events occurs on the client after RoboInst is installed:

1. RoboInst writes the network location of the miniroot (and a new IRIX 6.5 *sash* if the client is running an earlier release of IRIX than 6.5) in the disk volume header on the clients.
2. RoboInst reboots the client using the new *sash*.
3. RoboInst relocates the swap and root partitions on the client disk if a relocation is specified (see “Disk Partitions and RoboInst,” below).
4. The *sash* sets an environment variable on the client to signal a RoboInst installation; then it loads the miniroot from its network location to the swap partition on the client.
5. The miniroot configures networking via DHCP, BOOTP1533, or the PROM variables, as specified in the *startup* command.
6. The miniroot copies the files in the configuration directory to the */custom* directory in the miniroot filesystem.
7. Actions specified in the *mrconfig* file (such as *fx*, *inst*, and *postinst* commands) are executed.
8. The client system is returned to multiuser mode.

Disk Partitions and RoboInst

As Figure 9-1 illustrates, the physical location of the swap and root partitions on IRIX 6.5 systems are reversed from their traditional location in earlier IRIX versions. This relocation of *swap* and *root* permits arbitrary disk repartitioning during miniroot installations. When you upgrade a client system from an earlier IRIX version to IRIX 6.5, you can reverse *swap* and *root* automatically with an argument to the RoboInst command or with a configuration script (see “Launching RoboInst” on page 162). This disk reconfiguration is not necessary unless you plan to do disk partitioning from the miniroot.

Note: Although the location of swap and root are reversed, their partition numbers remain the same (root remains partition 0 and swap remains partition 1).

Getting Ready for RoboInst

This list suggests what to do to prepare for using RoboInst for miniroot installations. The subsections that follow the list provide details for each task:

1. Identify the client systems to be installed. See “The Client System List” on page 151.
2. Select and prepare the servers. See “Software Distribution Server Setup” on page 151, “Boot Server Setup” on page 152, and “Configuration Server Setup” on page 152.
3. Create the configuration directory. See “Configuration Directory Setup” on page 153.
4. Configure DHCP if you plan to use it. See “Network Setup” on page 153.

The Client System List

To prepare for RoboInst installations, create a list of hostnames of the client systems for which a miniroot installation is needed. These hostnames will be used in the RoboInst command to launch the installations (see “Launching RoboInst” on page 162).

In addition to hostnames, you should also make a list of the different system models of your clients. Use the list to verify that the boot server contains the boot file that is necessary for each system model (see Figure 9-1).

Software Distribution Server Setup

The software distribution server (or *installation* server) makes available the software images that Inst installs on other hosts. The images may be stored in directories on the hard disk or on CDs. If you plan to store the software distribution on a hard disk, plan on allocating approximately 0.5 GB for each CD in the original distribution. If you prefer, you can continue to use CDs as the software distribution source.

Instructions for configuring an installation server are given in Chapter 2 (see “Setting Up an Installation Server” on page 16 for details). If you follow those instructions to set up the installation server for miniroot installations (this means enabling BOOTP and TFTP on the server), you can use the same host as the installation server, the configuration server, and the boot server. (See also “Enabling BOOTP Forwarding on Routers” on page 16 and “Enabling TFTP Access on an Installation Server” on page 17.)

Boot Server Setup

The boot server contains *sash* and miniroot boot files for each model of client systems that it serves. The total disk space required for miniroot files is approximately 50 to 100 MB, depending on the number of models represented in your client set. If your network contains subnets, installations will require less time if an installation server is located on each subnet where clients are located; and locating the boot server on the same subnet saves additional time.

You will find the sash file, *sa*, in the *dist* directory of the Installation Tools distribution—on CD or provided by a software installation server. Various miniroot kernels are stored in the *miniroot* subdirectory of the Installation distribution. You can run *hinv* on the clients to determine their IP number to install the correct kernel on each. You should copy the *sa* file, and the miniroot subdirectory into a directory on the boot server (see “Example Boot and Configuration Server” on page 160 for an example.)

The server must be running BOOTP and TFTP to boot clients remotely. BOOTP is required to respond to client boot requests, and TFTP is required to transfer the boot files after the request is received. If you have not already done so, follow the instructions in “Enabling BOOTP Forwarding on Routers” on page 16 and “Enabling TFTP Access on an Installation Server” on page 17 to set up BOOTP and TFTP.

Note: If you combine the boot server with an installation server that is configured for miniroot installations, BOOTP and TFTP are already configured. See “Setting Up an Installation Server” on page 16.

Configuration Server Setup

The configuration server contains miniroot configuration files in a special configuration directory. Because they occupy miniroot space during the installation process, the files in this directory are necessarily small; therefore, disk space is seldom a consideration when selecting a configuration server.

If you decide to create the configuration server and boot server on separate hosts (or are launching RoboInst from the PROM monitor as described in “Launching RoboInst From the PROM” on page 163), be sure to make a note of the configuration server’s IP address. When the configuration server and boot server are on different hosts, you must specify the configuration server’s IP address when launching RoboInst (see “Launching RoboInst” on page 162).

The configuration server must permit access to files in the configuration directory. To allow this access, you can either enable TFTP on the configuration server if you have not already done so (see “Enabling TFTP Access on an Installation Server” on page 17 for instructions); or, if you prefer, you can use an open guest account instead of TFTP (see “Configuring an Installation Account” on page 18 for setup instructions).

Configuration Directory Setup

By convention, the configuration directory is `/usr/local/boot/roboinst/custom` on the configuration server. However, you can use a different directory if you specify the alternative name in RoboInst commands (see “Launching RoboInst” on page 162). After you create the configuration directories, populate them with an `mrconfig` file (see “Creating a Miniroot Configuration File” on page 154). You can also create and copy additional configuration files to control events before, during, or after the software installation process (see “Creating Additional Configuration Files” on page 160). After creating configuration files, run the `roboinst_config` command to generate a `.index` file. This file is used to confirm that all files are copied successfully to the client.

Network Setup

After RoboInst loads the miniroot, it checks the environment to determine which of the following three ways to start networking:

1. (default) RoboInst uses `netaddr`, the PROM value for IP address, to make an extended BOOTP (RFC 1533) request for network parameters (that is, netmask, gateway(s), static route(s), and so on).
2. RoboInst makes a Dynamic Host Configuration Protocol (DHCP) request for an IP address, hostname, and netmask parameters. If the responding server offers DHCP service, the server returns the requested parameter information.
3. RoboInst does not ask the network for information, but rather uses the PROM values for IP address and netmask (if set). This is a fallback procedure if #1 and #2 (above) should fail.

If you prefer, you can create a custom script to set the client’s IP address and other operating parameters (see the `nvr(1M)` reference page).

Customizing Miniroot Installations

To customize a miniroot installation, prepare a miniroot configuration file and any additional files that you want to execute as part of the installation session. If you use TFTP on the configuration server and the configuration directory contains additional files, it must also contain an index file that lists each file in the configuration directory.

Creating a Miniroot Configuration File

The miniroot configuration file, *mrconfig*, controls RoboInst installation sessions. This is an ASCII file in the configuration directory that contains keywords that RoboInst interprets to partition disks, install software, and so on, from the miniroot of the client system. Example configuration files that you can study and modify are located in */usr/share/src/RoboInst* (if you have installed the examples from the RoboInst distribution, they are not installed by default).

Each significant line of an *mrconfig* file begins with a keyword; blank lines and comments (lines beginning with the # character) are ignored. Keywords and arguments are described in Table 9-1.

Table 9-1 Keywords in an *mrconfig* File

Keyword	Arguments	Description
loghost	IP address (or hostname if boot server)	Specifies the host where log messages are sent in addition to the client system.
partition	device size type name options	Specifies partitions to create and filesystems to mount.
setenv	variable, value	Sets the named variable in the RoboInst environment. This variable is also exported to subcommands, such as those executed before, during, and after the software installation phase.
disksetup	None	Reverses the location of the swap and root partitions on a disk that was partitioned before IRIX 6.5. CAUTION: Any data contained on the disk is lost when you use this keyword.

Table 9-1 (continued) Keywords in an `mrconfig` File

Keyword	Arguments	Description
<code>init</code>	<code>/bin/sh</code> command	Specifies shell commands to execute during the initialization phase of the installation.
<code>fx</code>	<code>/bin/sh</code> command	Specifies shell commands to execute during the disk partitioning phase (<i>fx</i>) of the installation (see the <i>fx</i> manual page).
<code>mkfs</code>	<code>/bin/sh</code> command	Specifies shell commands to execute during the filesystem creation phase (<i>mkfs</i>) of the installation (see the <i>mkfs</i> manual page).
<code>preinst</code>	<code>/bin/sh</code> command	Specifies shell commands to execute during the phase immediately preceding software installation (<i>inst</i> execution).
<code>inst</code>	<code>inst</code> command	Specifies a command recognized by the <i>inst</i> utility. Each <i>inst</i> command must be preceded by this keyword. Commands are collected in the order that they appear in the <i>mrconfig</i> file, copied to a temporary file, and issued to <i>inst</i> with this syntax: <i>inst -r/root -c tempfile</i>
<code>onerror</code>	wait, ignore	If set to wait, suspend the installation and display the <code>Inst></code> prompt if an error occurs during the software installation process. If set to ignore, continue past error (still reported in <code>INSTLOG</code> and <code>SYSLOG</code>).
<code>nokernel</code>	None	Causes the autoconfiguration phase to be skipped so the IRIX kernel is not generated. If you use this keyword, you must create a script to build the IRIX kernel.
<code>postinst</code>	<code>/bin/sh</code> command	Specifies shell commands to be executed during the phase following software installation (<i>inst</i> execution).

Table 9-1 (continued) Keywords in an mrconfig File

Keyword	Arguments	Description
if ... endif	/bin/sh command	Similar to the if syntax in <i>/bin/sh</i> scripts. When an executed command exits with a status of zero, all lines up to the matching endif statement are evaluated. When an executed command exits with a nonzero status, the inclusion lines are ignored completely. Nesting is permitted.
if ... else ... endif	/bin/sh command	Similar to the if syntax in <i>/bin/sh</i> scripts. When an executed command exits with a status of zero, all lines up to the matching else statement are evaluated. When an executed command exits with a nonzero status, the lines between the else and endif statements are evaluated instead. Nesting is permitted.

Ordering Commands in an mrconfig File

Commands are parsed based on the keywords in the following order from the *mrconfig* file:

1. **loghost**
2. **init**
3. **partition**
4. **fx**
5. **mkfs**
6. **preinst**
7. **inst**
8. **nokernel**
9. **postinst**

While the convention of placing the commands in the *mrconfig* file in this order helps make it easy for users to read, it is not necessary for RoboInst.

Partition Keyword

Use the **partition** keyword to specify disk partitioning, filesystems and mount points. The syntax is as follows:

```
partition device size type name options
```

The *device*, *size*, and *type* arguments are used as input to the *fx -s* command (see the *fx(1M)* reference page) to perform the partitioning. Partitions are laid-out in the order specified, unless a start position is given.

device is a device name referring to a disk device in the */dev/rdisk* directory, for example *dks0d1s0*. See *dks(7M)* for a full description of device names.

size is one of the following arguments:

- **existing**—keep the same size.
- **standard**—use a standard layout for the entire disk. Type **root** or **option** arguments must also be specified. When **standard** is used, it applies to the entire disk (the partition component in the device name is irrelevant). For most partitioning tasks, either **standard root** or **standard option** should be specified as the first partition statement for the given disk. If customizations are desired, additional partition statements for that disk may also be specified.
- **all**—The entire disk.
- *start:size*

start is an integer that specifies the exact start address of the partition, in 512-byte blocks. Alternatively, *start* can be **followspart#** where # is the partition-number that this partition should immediately follow on the same disk. Note that the volume header is partition 8, so to use the first usable partition, you would normally use **followspart8**.

size is an integer that specifies the size of the partition in 512-byte blocks or the word **remainder** to use the entire remainder of the disk after making all of its other partitions. Note that partitions are processed in the order they appear in the file, so **remainder** should be used only in the last partition statement for a particular disk.

type is one of the following arguments:

- **root**—Valid only when **standard** is specified. A standard root disk is created, containing a swap partition and a root partition of maximum size containing an XFS filesystem.

- **option**—Valid only when **standard** is specified. A standard option disk is created, consisting of a single partition of maximum size, containing an XFS filesystem.
- **xfs** or **xfs/blocksize**
An XFS filesystem is created with the specified *blocksize*, and mounted at the directory *name* with the specified *options*. The blocksize must be an integer multiple of 512 and cannot exceed 65536. If omitted, a blocksize of 4096 is assumed.
- **efs**—An EFS filesystem is created and mounted at the directory *name* with the specified *options*.
Note: EFS filesystems will not be supported in future IRIX releases. Use XFS filesystems in nearly all situations.
- **swap**—A swap partition of the specified *size* is created.
- **preserve**—Any existing filesystem is preserved, and no new one is created.

name is ignored when *type* is **swap**. For other partition types, its value can be either of these:

- *pathname*—indicating a local directory where the filesystem is to be mounted.
- **nomount**—If **nomount** is specified, the filesystem is not mounted.

options are options to the *mount -o* command. This field is optional. Any options specified are to be passed along to the *mount* command as a single argument. Multiple options should be comma separated with no spaces. Refer to the mount(1M) reference page for more information.

For example, use this command to make disk (0,1) a standard system drive:

```
partition dks0d1s0 standard root /
```

For example, use this to make disk (0,2) a standard option drive mounted at */d2*:

```
partition dks0d2s0 standard option /d2
```

Hostname Considerations

No name services, such as DNS or UNS, are available from the miniroot. The only source of hostnames is the */etc/hosts* file and the hostname of the boot server. For this reason, all other remote host designations in the *mrconfig* file must be in the form of IP addresses, (unless you enter hostnames in the local */etc/hosts* from a script, for example).

You can add hostnames to the miniroot's */etc/hosts* file individually, for example:

```
init echo "192.1.1.2.3 fred1.acme.com fred1" >> /etc/hosts
```

You can also copy in a host file from the configuration directory, for example:

```
init cat /custom/hosts >> /etc/hosts
```

Environment Variables

RoboInst software sets the following variables in the environment of all shell commands defined in an *mrconfig* file:

SGI_ABI
SGI_BOOTDIR
SGI_BOOTSERVER
SGI_CAPACITY
SGI_CAP_dks#d#vol
SGI_CONFIGDIR
SGI_CONFIGSERVER
SGI_CPUARCH
SGI_CPUBOARD
SGI_CUSTOM
SGI_GFXBOARD
SGI_HOSTNAME
SGI_IPADDR
SGI_MACHINE
SGI_MEMSIZE
SGI_MODE
SGI_ROOT
SGI_SUBGR
SGI_SYSID

Refer to the `roboinst_config(1M)` reference page for details and the latest list of environment variables.

Creating Additional Configuration Files

If you use TFTP on the configuration server and the configuration directory contains files in addition to the `mrconfig` file, the configuration directory must also contain a `.index` file that lists each additional file. The `/usr/etc/roboinst_config` script, which is part of RoboInst software, creates a `.index` file automatically. When you have finished creating the configuration directory and all necessary files (or have just modified a file), run `roboinst_config` to create a new `.index` file (refer to the `roboinst_config(1M)` reference page for details).

The index file that `roboinst_config` creates lists the name, type, size, and checksum of each file in the configuration directory. RoboInst compares this information to verify that the configuration files copied to the client are identical to the files contained in the configuration server. If the files are not identical, they are not executed and the installation fails. For this reason, you should generate a `.index` file even if the configuration directory contains only an `mrconfig` file.

Example Boot and Configuration Server

Figure 9-2 shows an example of how a setup might look if your boot server and configuration server are the same host. A simple setup is shown in which the only configuration file is `mrconfig`, and the only boot files are the miniroot kernel, `unix.IP22`, and the sash, `sa`.

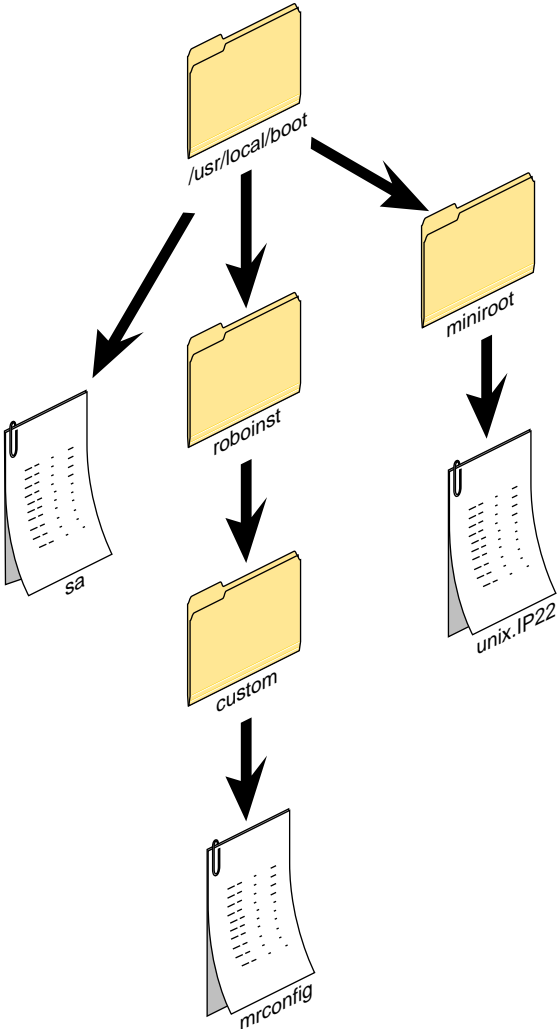


Figure 9-2 Sample RoboInst Directory and File Setup

Launching RoboInst

Launch RoboInst from a network host on which you have installed the RoboInst server software (refer to your release notes for client and server installation information). The command to launch RoboInst is *roboinst*, and basic command syntax is as follows:

```
roboinst [-n|-y] -b bootdir -c configdir -t time client(s)...
```

(Refer to the discussion below and the roboinst(1M) reference page for more command-line options and further details.)

Launching RoboInst From the Server

You can launch RoboInst from any supported IRIX server with the *roboinst* server software installed. You must specify a boot server (**-b** option) and a configuration server (**-c** option). You may specify a time (**-t** option) if *at* is enabled on the server, or else accept the default time “now.” You may also specify automatic installation (**-y** option), or just check the configuration without initiating installation (**-n** option). If you specify neither **-y** or **-n** you will be prompted before starting the installation.

In Example 9-1, a configuration test is made with the **-n** option, and an error is returned because the *mrconfig* file is not readable.

Example 9-1 RoboInst Configuration Test

```
bar1 1# roboinst -n -b fred1:/usr/local/boot \  
-c fred1:/usr/local/boot/roboinst/custom foo1 foo2 foo3 foo4  
roboinst: submitting job on bar1  
bar1: Size mismatch for /tmp/d.roboinst.3106/mrconfig 0 (expected 1719)  
bar1: Try running roboinst_config  
bar1: Unable to retrieve configuration from  
192.1.2.3:/usr/local/boot/roboinst/custom  
bar1 2#
```

In Example 9-2, RoboInst proceeds automatically, stopping at an error if *onerror* is set to *wait* (see Table 9-1).

Example 9-2 RoboInst Configuration Launch With Prompting

```
bar1 1# roboinst -b fred1:/usr/local/boot \  
-c fred1:/usr/local/boot/roboinst/custom foo1 foo2 foo3 foo4
```


Launching RoboInst from the Client

To launch RoboInst from a client, install server and client software on the host and then run the *roboinst* command as with a server. If you are only installing on the client you run the *roboinst* command from, you do not specify the client name, as shown in Example 9-3.

Example 9-3 RoboInst Installation From the Client

```
bar1 1# roboinst -b fred1:/usr/local/boot \
-c fred1:/usr/local/boot/roboinst/custom
```

Launching RoboInst From the PROM

If the client is not up and running on the network, you can initiate RoboInst from the PROM monitor of the client using a *boot* command with the argument *mrmode=custom*, as described in this section.

Caution: Pay special attention to the **disksetup** keyword. Use the **disksetup** keyword only if you plan to repartition the disk drive, because it causes all current contents of the root drive to be lost. (Subsequent to its first use, however, *disksetup* may not make any noticeable changes. In particular, it will not “wipe” the disk. Rather, it rearranges partitions, causing them (if changed) to not appear to have filesystems. In fact, if they have already been rearranged, *disksetup* may appear to do nothing.)

The final argument, *mrconfig=addr:/pathname* is used to specify the IP address and pathname of the directory containing your *mrconfig* file. For example:

```
boot -f bootp()server:/path/sa(sashARCS) mrmode=custom disksetup=true
mrconfig=130.62.51.86:/var/tmp/roboinst

boot -f bootp()server:/path/sa(sashARCS) mrmode=custom
mrconfig=130.62.51.86:/var/tmp/roboinst

boot -f bootp()server:/path/sa(sashARCS) mrmode=custom

boot -f bootp()server:/path/sa(sash64) mrmode=custom
mrconfig=130.62.51.86:/var/tmp/roboinst

boot -f bootp()server:/path/sa(sash64) mrmode=custom
```

In the examples above, it is necessary to use an IP address for the configuration server only if it is different from the boot server. If these two servers are the same, then a hostname can be specified:

```
boot -f bootp()server:/path/sa(sashARCS) mrmode=custom
mrconfig=server:/var/tmp/roboinst
```

Note: The PROM has a limit on the command line length. If you need to, you can work around this limit in any of several ways:

1. User a shorter directory path on the server(s).
2. Separate the command into multiple commands. For example, you can set most of the variables as separate commands:

```
setenv mrconfig 130.62.51.86:/var/tmp/roboinst
boot -f bootp()server:/path/sa(sashARCS) mrmode=custom \ disksetup=true
```

3. Configure DHCP or BOOTP1533 to return the *mrconfig* path in the *pro_extensions_pathname* (proclaim configuration syntax) option. In this case, the file returned would contain the following:

```
pro_roboinstdir=server-IP:/path/to/mrconfig/dir
```

4. Use the default *mrconfig* paths (in the following order):

```
dhcpserver:/usr/local/boot/roboinst/custom
bootserver:/usr/local/boot/roboinst/custom
```

Note: The *boot* command may fail if you have not properly set your *netaddr* variable. Use this PROM command to display the current value:

```
printenv netaddr
netaddr=130.62.51.201
```

If it is not set correctly to the IP address of the host, use the *setenv* command to set it correctly, for example:

```
>> setenv netaddr 143.69.51.201
```

Contact your network administrator if you are not sure what the appropriate address is. On some networks, if you have a properly configured bootp server with your machine listed in its */etc/bootptab* file, your *netaddr* variable will be set automatically if you first unset it with this command:

```
unsetenv netaddr
```

Then use the *boot* command.

Additional RoboInst Command-Line Options

A useful command option to know is the *roboinst -L* “live mode” option. With this option, RoboInst does not bring the client down to the miniroot to perform the automatic installation. Only the **preinst**, **inst** and **postinst** keywords in the *mrconfig* file are processed with this option, and it is not possible to perform disk reformatting or modify networking parameters in live mode.

Another useful command to consider is *roboinst -f*. With the *-f* option, you can automatically install the roboinst client software on the clients.

Refer to the *roboinst(1M)* reference page for more information on these and other options.

PART THREE

Licensing

Chapter 10, "Licensing"

Licensing

Silicon Graphics products are licensed by the FLEXlm application. To set up and maintain the licensing scheme for these products at your site, use the *FLEXlm End User Manual* (available online through IRIS InSight) as your primary source of license administration information. The InSight version of the FLEXlm guide contains icons that identify information that either does not apply to the Silicon Graphics implementation of FLEXlm or requires additional explanation. This chapter of *IRIX Admin: Software Installation and Licensing* provides the alternative and additional information that is marked by the icons.

This chapter contains these sections:

- “About License Manager” on page 169
- “The License Request Process Illustrated” on page 171
- “FLEXlm Files and Utilities on Silicon Graphics Systems” on page 172
- “New FLEXlm Features” on page 174
- “Starting and Stopping Licensing Daemons” on page 176

Note: Some older applications running on Silicon Graphics systems use NetLS licensing; please see the *Network License System Administration Guide* for information on managing these licenses.

About License Manager

Users license their applications with License Manager, a graphical tool that can be accessed from the System toolchest. The License Manager dialog is shown in Figure 10-1. Whenever a user installs, updates, or removes a floating licence with this dialog, License Manager restarts (or stops, for license removals) the local license manager daemon to put the user’s change into effect.

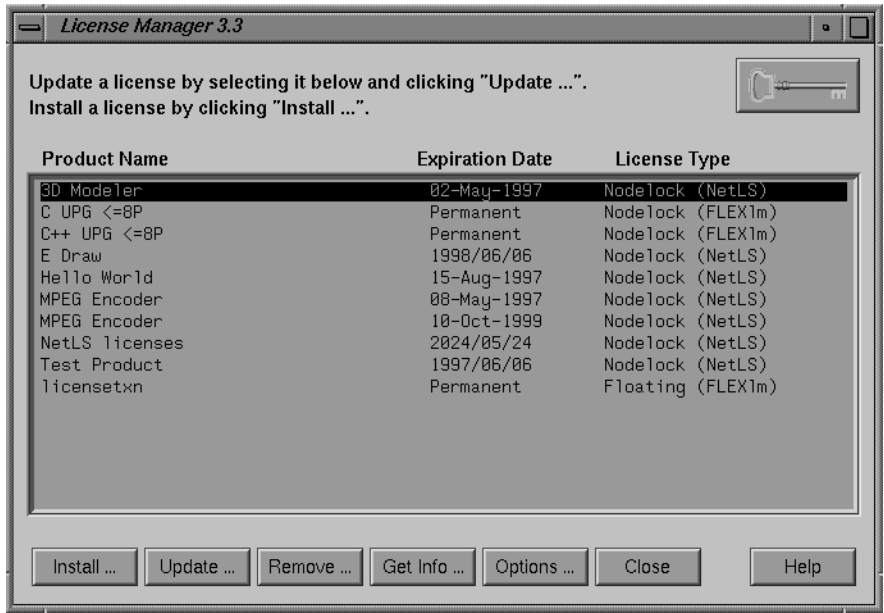


Figure 10-1 License Manager Dialog

Starting with IRIX 6.5, the License Manager dialog contains an *Options* button. This button allows users to manage licenses for Alias/Wavefront applications in addition to the group of applications that are normally licensed on Silicon Graphics systems.

See the LicenseManager(1M) reference page for more information on the user interface to licensing.

License Types

The FLEX1m implementation on Silicon Graphics systems supports these license types:

- A node-locked license permits a particular system to run an application. The number of users is not restricted with this type of license.
- Floating licenses permit a fixed number of users on a network to use an application simultaneously.

- Incremental licenses permit the number of users on a floating license to be increased. The total number of users permitted to use an application is the sum of users specified on the INCREMENT lines for an application listed in the floating license file (see “FLEXlm Files and Utilities on Silicon Graphics Systems” on page 172).
- Package licenses permit several applications to use a single license. The license is listed in a PACKAGE line in the floating license file. This line contains a COMPONENTS attribute that identifies the applications that the package licenses (see “FLEXlm Files and Utilities on Silicon Graphics Systems” on page 172).

The License Request Process Illustrated

Figure 10-2 illustrates the events in the license request process.

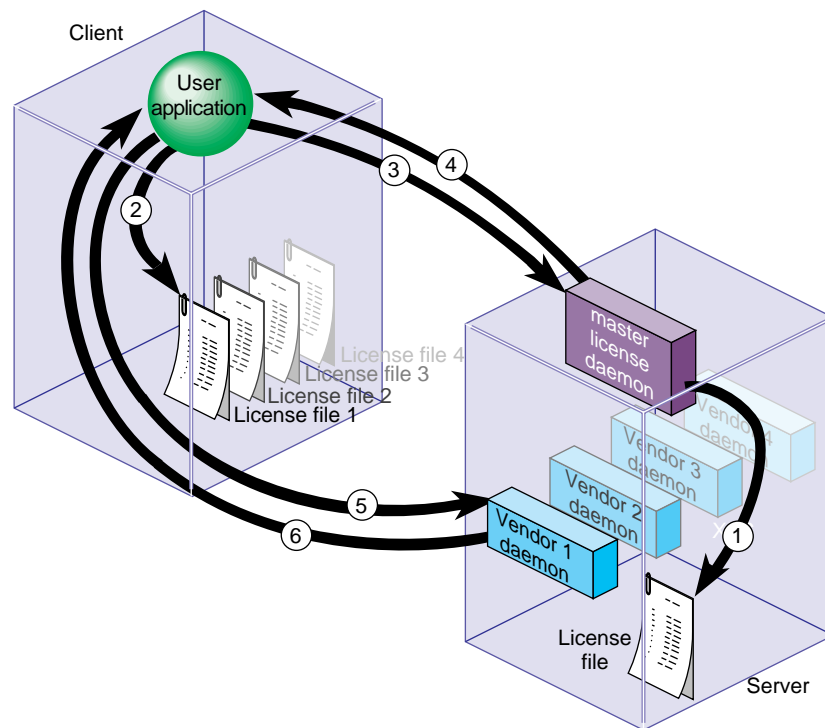


Figure 10-2 License Request Process

The license request process consists of this sequence of events:

1. When it starts, the master license daemon (*lmgrd*) reads the server copy of the license file and starts the vendor daemons listed in the file.
2. When it starts, the user application searches the client copy of the license file for the address of the license server, which includes the port number for *lmgrd*.
3. The user application requests the vendor daemon's address from the master daemon.
4. The master daemon sends the address of the vendor daemon to the user application.
5. The user application requests a license from the vendor daemon, using the address it received.
6. The vendor daemon checks its memory for available licenses and grants or denies a license.

FLEXlm Files and Utilities on Silicon Graphics Systems

When you install FLEXlm software on Silicon Graphics systems, licensing files and programs are stored in these locations:

- /usr/etc/lmgrd* license manager daemon
- /usr/etc/sgifd* Silicon Graphics vendor daemon
- /var/flexlm/licensefile.db*
database of license files for Silicon Graphics applications
- /var/flexlm/license.dat*
license file for node-locked licenses
- /var/flexlm/license_<hostname>.dat*
license file for floating licenses on the *hostname* server
- /usr/sbin (lmdiag, lmdown, and so on)*
FLEXlm license administration tools
- /etc/init.d/flexlm*
Startup script for the FLEXlm license daemon

/etc/config/lmgrd.options
Configuration file containing arguments to the *lmgrd* command

/etc/config/flexlm
Text file for enabling or disabling automatic startup of the FLEXlm daemon

Licenses for Alias/Wavefront applications are stored in separate files:

/var/flexlm/aw.dat
license file for node-locked licenses to Alias/Wavefront applications

/var/flexlm/aw_<hostname>.dat
license file for floating licenses to Alias/Wavefront applications on the *hostname* server

The License File Database for Silicon Graphics Applications

To locate their licenses, Silicon Graphics applications search the */var/flexlm/licensefile.db* file, which is located on the client system and contains a list of license files and directories. These applications search each file listed in *licensefile.db* whose name ends in *.dat*; they also search all *.dat* files stored in *licensefile.db* directories. The contents of */var/flexlm/licensefile.db* are illustrated in Example 10-1, which contains one license directory and two license files:

Example 10-1 Database of License Locations

```
/var/flexlm  
/var/flexlm/license.dat  
/usr/lib/SoftWindows/FLEXlm/license.dat
```

A license file in */var/flexlm/licensefile.db* might direct the application to a license server to continue its search for a license (see “The USE_SERVER Feature” on page 174 for details). If you plan to store licenses in a directory or filename that is different from those specified in */var/flexlm/licensefile.db*, or if you create new locations for license files, you must modify */var/flexlm/licensefile.db* to reflect your changes.

Note: The license daemon does not search */var/flexlm/licensefile.db* to determine license file locations; *licensefile.db* is used only by Silicon Graphics applications. See “Specifying License File Locations,” which follows.

Specifying License File Locations

Neither the license manager daemon (*lmgrd*) nor the vendor daemons use */var/flexlm/licensefile.db* to determine license file locations. These daemons locate license files by other methods:

- On Silicon graphics servers, the master license daemon is started with parameters specified in the */etc/config/lmgrd.options* file. The *-c* argument in this file specifies the location of the license file. If you change the location of the license file, you must modify */etc/config/lmgrd.options* to specify the new license file location.
- Applications whose license files are not listed in */var/flexlm/licensefile.db* locate license files by means of the *LM_LICENSE_FILE* environment variable (also see “The port@host Feature” on page 175). To specify the location of license files, enter a line like the sample below in the *.cshrc* file (*.profile* or *.kshrc* for Bourne or Korn shells) on the client system:

```
LM_LICENSE_FILE=<licensefile_path1>:<licensefile_path2>...
```

New FLEXlm Features

The FLEXlm application contains two new features that can be implemented with applications using FLEXlm version 5 (or later): USE_SERVER and an enhancement to the port@host feature.

The USE_SERVER Feature

Version 5 of the FLEXlm application offers USE_SERVER, which eliminates the need to copy and maintain the server’s license file on client systems. With this feature, only two lines are required in the client’s license file: a SERVER line and the USE_SERVER line. When an application reads the USE_SERVER line, it reads the license file on the specified server instead of reading the local file.

Example 10-2 illustrates a client license file that implements the USE_SERVER feature. In this example, the application is directed to server sparkplug to read its license file:

Example 10-2 Client License File With the USE_SERVER Feature

```
SERVER sparkplug 023456 1701
USE_SERVER
```

Note: The USE_SERVER feature is valid only for applications built with FLEXlm version 5.

The port@host Feature

The version 5 enhancement to the port@host specification provides the same function as the USE_SERVER feature: it directs applications to read their license file from a remote server instead of reading it locally. Like earlier versions of port@host, the enhanced port@host uses the *LM_LICENSE_FILE* environment variable to specify the server. For example, the specification in Example 10-3 directs applications to read the license from sparkplug by means of a connection to port 1701.

Example 10-3 Specification for the Enhanced port@host Feature

```
LM_LICENSE_FILE=1701@sparkplug
```

It is possible to specify unenhanced port@host processing (FLEXlm version 4) for applications built with FLEXlm version 5. To specify unenhanced port@host processing, prepend a minus sign (-) to the LM_LICENSE specifications (shown in Example 10-4).

Example 10-4 Specification for Unenhanced Processing of port@host

```
LM_LICENSE_FILE=-1701@sparkplug
```

Starting and Stopping Licensing Daemons

The installation process installs a startup script, `/etc/init.d/flexlm`, that starts and stops the FLEXlm license daemon. This script executes the `lmgrd` command to start the license daemon, using the values in the configuration file `/etc/config/lmgrd.options`.

In its startup sequence, the system reads the file `/etc/config/flexlm` to determine whether to start the license daemon. By default, this file contains the value `off`, which prevents the daemon from starting. To enable the `lmgrd` daemon, change the value in `/etc/config/flexlm` to `on` using this commands:

```
# /etc/chkconfig flexlm on
```

To start the FLEXlm license daemon, enter this command as superuser:

```
# /etc/init.d/flexlm start
```

To stop the FLEXlm license daemon when it is running, enter this command:

```
# /etc/init.d/flexlm stop
```

Note: The FLEXlm daemon is started and stopped automatically when a workstation user installs, updates, or removes a license with the License Manager (see “About License Manager” on page 169).

Inst Quick Reference

This appendix is a reference for experienced Inst users. It contains an abbreviated installation procedure and a complete glossary of Inst commands. The command glossary includes hidden commands and provides expanded discussions of command features that are partially covered in earlier chapters. Commands on each menu are presented in alphabetic order.

This chapter contains these sections:

- “Main Menu Commands” on page 180
- “View Menu Commands” on page 187
- “Administrative Menu Commands” on page 189
- “Interrupt Menu Commands” on page 192

Condensed Installation Instructions

Use this procedure if you are already familiar with Inst and need only cursory instructions to complete the installation process. The procedure contains references to information in Chapter 5, in case you need more detail.

Note: Inst has an extensive online *help* feature. To see the list of topics on which help is available, give the command *help help*.

1. Become the superuser.

```
% su -  
Password:
```

2. Invoke Inst:

```
# inst
```

After your entry, initial messages appear, including the default location of the software distribution that will be used for the installation. If you plan to use this distribution, omit step 3 and go directly to step 4.

3. Specify the software distribution source, if necessary:

```
Inst> from server:pathname
```

Use the *from* command to specify the location of the distribution that you wish to use if the default is not acceptable. Substitute a local specification, such as */CDROM/dist*, if necessary. (See “Step 2: Specifying the Source” on page 41 for more details.)

The initialization is performed after your entry. Inst posts messages about reading the installation history and product descriptions; then it performs space calculations. Note that if you issue the *from* command, any currently open distribution is closed and any current selections are cleared.

4. Open additional distribution sources, if necessary:

```
Inst> openserver:pathname
```

As of IRIX 6.5, multiple distributions may be opened simultaneously. The *open* command is identical to the *from* command, except that *open* will open one or more additional software distributions without closing the current distribution, and without clearing the selections you have already made.

5. Display the distribution inventory (optional):

```
Inst> list
```

Use the *list* command to browse the software on the distribution. After your entry, you see a display of subsystems, their selection status, installation status, and disk space requirements. (See “Using the list Command” on page 43 for details.) If you omitted step 3, you see the initialization messages before the software inventory is posted.

6. Switch to the target view and look at the installed products (optional):

```
Inst> view targ
```

```
Inst> list
```

To browse the software on the target, use *view* to change the view to the target and *list* to display target software. After your entry, you see displayed subsystems that are installed on the target and their removal status. (See “Using the View Commands Menu” on page 64 for details.)

7. Step through the distribution inventory to change selections, as necessary:

```
Inst> view dist
Inst> step
```

After your entries, the first subsystem in the distribution is displayed. Type `<Enter>` to display succeeding subsystem, checking the selection status in the first column of each subsystem as it is displayed (these selections were made by Inst during initialization). Use the selection commands to change any selections, as needed: Type *i* to install the subsystem, *r* to remove it, and *k* to keep the subsystem in its present condition on the target. (See “Step 4: Making Software Selections” on page 48 and “Using step to Make Selections” on page 90 for more information.)

8. Launch the installation:

```
Inst> go
```

If there are no conflicts or errors in your selections, see messages about images being removed and installed, followed by a completion message, after you enter *go*. (See “Step 5: Launching the Installation” on page 51 for details.) If you see conflict messages, go on to step 8 of this procedure. If no conflict messages are posted, omit steps 8 and 9 and go directly to step 10.

9. Resolve conflicts, if necessary:

```
Inst> conflicts num num num
```

If you were notified of conflicts, specify the conflict resolution that you prefer from the list of recommendations. (See “Step 6: Resolving Conflicts” on page 52 for details.)

If you prefer not to accept the resolutions that are offered in the conflict notice, you can return to step 6 to change your selections. Then repeat the *go* command to install or remove the software that is affected by the conflict resolutions.

10. Remove some selections if there is insufficient disk space.

If you were notified that your selections required more disk space than Inst can safely use, you must remove some of your selections. Repeat the *go* command to process your changes.

11. Quit Inst.

If there is no more software to install, quit Inst at the quit prompt (see “Step 7: Quitting the Installation Session” on page 55 for details). Otherwise, return to step 3 to change the distribution source or step 4 to display the distribution inventory.

Main Menu Commands

The list below describes all commands (including hidden commands) that appear on the Inst Main menu.

admin Use the *admin* command to display the Administrative Commands menu. This provides a variety of functions that are not needed for most installations.

close Use the *close* command to close a software distribution that has been previously opened with the *from* or *open* commands. Enter the *close* command with no arguments to get a list of open distributions and prompting to close them.

conflicts Use the *conflicts* command with no arguments to display the list of conflicts that Inst has detected in your installation selections. Conflicts occur if you select incompatible software for installation, if you omit required selections, or if you omit subsystems that are prerequisites to other selected subsystems.

If a conflict occurs, the *go* command is not processed; an error message is posted that includes a list of suggested resolutions. Use the *conflicts* command to specify your choice of conflict resolutions in the conflicts message:

```
Inst> conflicts 1a 2b 3b
```

Inst uses conflict choices to modify your original selections and complete the installation. If you prefer, you can modify your selections with *install*, *remove*, and *keep* commands instead of accepting the suggested conflict resolutions.

from Use the *from* command to specify the location of the software distribution that you wish to install. Your specification can include a single product in the distribution. The syntax of the argument to the *from* command is explained in “Alternative Ways to Specify a Distribution” on page 83.

The *from* command causes four events:

- A connection with the distribution source is established.
- The product descriptions are read from the source. Inst displays the list of product descriptions that it reads.

- The current selections for installation and removal are discarded. If the *set* preference *autoselect* is *on*, Inst uses its algorithms for making initial installation selections. If *autoselect* is *off*, nothing is selected for installation (see “The Preferences List” on page 128 for more information).
- If the *delayspacecheck* preference is *off*, disk space is calculated before the output of a *list* or *step* command is displayed (see “The Preferences List” on page 128 for more information).

If the distribution is on a remote server, Inst must have access to the IP address of the server. If the server address is not listed in the target’s */etc/hosts* file (*/etc/hosts* or */root/etc/hosts*), Inst prompts you for the address.

When Inst is invoked, it displays a message about the default location of the software distribution. If this location is correct, it is not necessary to enter the *from* command.

go Use the *go* command to start the installation and removal of files that you selected with *install*, *remove*, and *keep* commands. Before the installation begins, Inst does several checks to verify that your selections are valid. If any check fails, no installations or removals are performed and an error message is displayed.

help Use the *help* command to display information on a given topic:

```
Inst> help topic
```

The *topic* can be almost any word that appears on a menu, letters that appear in the output of *list* and *step*, and the special vocabulary of the Inst application.

If you are unfamiliar with Inst or feel confused, start with *help help*.

If *help* output is more than one screenful, a *more?* prompt appears. Press the Space Bar when you are ready to display more output.

A complete list of *topic* arguments to *help* is shown in Table A-1.

	Table A-1 Help Topics				
*	close	i	mount	readerror	space
.N	config	image	n	recalculate	spaceerror
.O	conflicting	incompatible	names	relocate	stale
?	conflicts	inst	networkerror	removable	standard
A	d	install	new	remove	step
C	date	installable	nfs	request	stream
D	default	installed	notinstalled	required	streams
I	deltas	k	open	return	subsystem
L	devices	keep	options	rqs	targ
N	dir	keepfile	overlay	s	target
P	dist	kernelerror	overlays	same	timeouterror
R	distribution	keywords	overview	save	u
S	downgrade	level	patch	selection	umount
U	files	list	patches	set	unmount
a	filesystems	load	patchupgrade	sethostipaddr	updatekeepfile
admin	filter	lverror	preferences	sethostname	upgrade
all	fresh	main	prereqs	sh	upgraded
args	from	maint	prerequisite	show	view
b	fsname	maintenance	product	shroot	wildcard
base	go	miniroot	q	sizes	
beginner	hardware	mkfs	quit	sort	
check	help	more	r	source	

- install* Use the *install* command to select products or product components for installation. The *install* command queues your request; the installation occurs only after you use the *go* command. Until you enter the *go* command, you can change your selections as you wish. Use the *keep* command to undo an *install* request.
- keep* Use the *keep* command to select products or product components that should be kept as is on the target. Also use the *keep* command to undo other requests—*keep* cancels any pending *install* or *remove* selections. The existing versions on the target, if any, remain undisturbed.
- list* Use the *list* command to show the products or product components that are in the currently available distribution. By default, *list* displays at the subsystem level. See “Interpreting list Output” on page 45 for a description of the output. Also see “Using the View Commands Menu” on page 64 for information on controlling *list* output.
- open* Use *open* to open another distribution without closing the current distribution. (See the *from* and *close* commands as well.)
- quit* Use *quit* to exit Inst.
- If you made installation or removal selections that were not processed by the *go* command, Inst reminds you of pending selections and asks you to confirm the *quit*. If you type *yes*, the selections are discarded. If you type *no*, you return to the current menu. You can use *list* to identify pending selections and *go* to install or remove them.
- Usually, auxiliary commands are executed automatically after you use *quit*. If you are installing software that requires a system reboot, you will be asked if you are ready to restart the system. If you answer *no*, the Inst session resumes. If you answer *yes*, the system reboots and resumes operation using the new software.
- remove* Use the *remove* command to select installed products or product components that you wish to remove from the target. The removal does not occur until you enter the *go* command. Until then, you can change your selections for any or all subsystems. To undo a *remove* request, use the *keep* command.
- Inst automatically removes old versions of software when it installs a upgrade to the software, so it is not necessary to remove old versions explicitly. Typically, software is removed when it is no longer needed or when the target system is short of disk space.

set Use the *set* command with no arguments to display a list of preferences that allow you to tailor Inst operation. To change a preference, use *set* with the preference and its value as arguments:

```
Inst> set delayspacecheck on
```

For a list of preferences, see “The Preferences List” on page 128.

sh Use *sh* with no arguments to invoke an IRIX shell. Use *sh* with an IRIX command as its argument to execute the IRIX command. This command displays the contents of the */usr/tmp* directory on the target:

```
Inst> sh ls /usr/tmp
```

See “The sh and shroot Commands” on page 75 for additional information.

shroot Use *shroot* with no arguments to invoke an IRIX shell that executes commands relative to a new root (see the *chroot(1M)* reference page). Use *shroot* with an IRIX command as an argument; the command is executed relative to the new root. This command displays the contents of */root/usr/tmp*:

```
Inst> shroot ls /usr/tmp
chrooting to /root
```

See “The sh and shroot Commands” on page 75 for additional information.

step Use the *step* command to display and select individual products or product components. The *step* command takes keyword arguments. It also takes the name of a product, image, or subsystem as an argument. See “Using step to Make Selections” on page 90 for additional information on *step*.

When you use *step*, Inst reads single keystrokes. The key that you press as each item is displayed determines the action that is taken.

Table A-2 describes the effects of keystrokes during *step* operations. The items in this table assume that the viewing level is subsystem (see “The level Command” on page 67 for information on setting viewing levels).

Table A-2 Keys Used in step Mode

Key	Action
d	Select this subsystem for installation if marked with <i>d</i>
f	List the files in this subsystem
h	Display help for the <i>step</i> feature
i	Select this subsystem for installation
j	Retain the current request and display the next subsystem
k	Keep the current version and/or cancel the current request
l	List all previous subsystems
n	Retain the current request
p	Retain the current request and display the previous subsystem
q	Quit step mode
r	Select this subsystem for removal
u	(Same as i)
D	Select the remaining subsystems in this product for installation if marked with <i>d</i>
I	Select the remaining subsystems in this product for installation
J	Retain the current request for all subsystems in this product and display the first subsystem in the next product
K	Keep the current version and/or cancel the current request for the remaining subsystems in this product
N	(Same as J)
P	Retain the current request and display the first subsystem of the previous product
R	Select the remaining subsystems in this product for removal

Table A-2 (continued) Keys Used in step Mode

Key	Action
U	(Same as I)
<Enter>	(Same as j or n)
+	(Same as j , n , or <Enter>)
-	(Same as p)
/pattern<Enter>	Search for pattern specified after the slash
<right-arrow>	(Same as i or u)
<left-arrow>	(Same as k)
<down-arrow>	(Same as n , j , + , or <Enter>)
<up-arrow>	(Same as p or -)
<Shift-right-arrow>	(Same as D)
<Shift-left-arrow>	(Same as k)
<Shift-down-arrow>	(Same as J or N)
<Shift-up-arrow>	(Same as P)

view Use the *view* command without arguments to display the View menu. Use the *view* command with arguments (View menu items) to control the output of the *list* and *step* commands. (See “Using the View Commands Menu” on page 64 for more information on *view* features.)

By default, *list* and *step* display all software in the distribution at the subsystem level; subsystems are sorted alphabetically by shortname. You can use *view* to change these defaults. For example, these commands set the view to the target and reset it to the distribution:

```
Inst> view targ
Inst> view dist
```

These commands specify the viewing level:

```
Inst> view level image
Inst> view targ
```


These commands change the sorting index to size and filter out downgraded distribution items from displayed lists:

```
Inst> view sort size
Inst> view filter downgrade off
```

Also use *view* commands with the *set* command to set Inst preferences (see “A Closer Look at Preferences” on page 126 for details).

View Menu Commands

The list below describes all commands that appear on the View Commands menu.

- | | |
|--------------------|--|
| <i>deltas</i> | Use the <i>deltas</i> command to show the size of subsystems relative to their installed counterparts, rather than showing their absolute sizes (in <i>list</i> output). Also see <i>sizes</i> . |
| <i>dist</i> | Use the <i>dist</i> command to set the view to the distribution. When the view is set to <i>dist</i> , the <i>list</i> and <i>step</i> commands display software in the distribution inventory. See also the <i>targ</i> command description in this section. |
| <i>filesystems</i> | Use the <i>filesystems</i> command to display local filesystems separately in <i>list</i> output. Also use the <i>filesystems</i> command to display NFS filesystems and read-only filesystems, and to display individual local filesystems, NFS filesystems, and read-only filesystems.

<pre>filesystems on filesystems nfs filesystems readonly filesystems all</pre> |
| <i>filter</i> | Use the <i>filter</i> command with no arguments to display the list of filters that can be used to modify the output of <i>list</i> and <i>step</i> commands. This form of the command also shows the current settings for the filters (on or off). By default, all filters are off.

Use the <i>filter</i> command with one of these arguments to modify the output of <i>list</i> and <i>step</i> commands:

<pre>new upgrade patchupgrade same downgrade notinstalled</pre> |

For example, this command causes new products to be omitted from *list* and *step* displays:

```
View> filter new off
```

- help* Use the *help* command to display Inst online help. Help is available on all topics listed in Table A-1.
- level* Use the *level* command with no arguments to display the current level of the software hierarchy (see Figure 1-1 on page 6) that is shown in *list* and *step* displays. By default, software is displayed at the subsystem level. Use the *level* command with one of these arguments to change the level of software shown in *list* and *step* displays.

```
product (or 1)
image (or 2)
subsystem (or 3)
```
- return* Use the *return* command to return to the Main menu.
- set* Use the *set* command to examine and change the settings of variables called preferences. If you provide no arguments, the current settings are displayed. If you provide the name and a value, the preference value is changed.

Most preference settings are saved in the installation history when you enter the *quit* command and are restored the next time that you invoke Inst. “The Preferences List” on page 128 describes *set* preferences in detail.
- sizes* Use the *sizes* command to show the absolute size of subsystems (the default view) rather than their size relative to an installed counterpart (in *list* output). Also see *deltas*.
- show* Use the *show* command to display the current view settings.
- sort* Use the *sort* command with no arguments to display the current sorting order of *list* and *step* displays. By default, software is sorted alphabetically by shortname, but you can use the *sort* command to change the sort order so that subsystems are shown in ascending order of size. After your entry, the new sort order is displayed:

```
View> sort size
Current Sort : size

View> sort name
Current Sort : short
```

targ Use the *targ* command to set the view to target software. When the view is set to *targ*, the *list* and *step* commands display software that is either installed on the target or recorded in the installation history. See also the *dist* command description in this section.

Administrative Menu Commands

The list below describes all commands (including hidden commands) that appear on the Administrative Commands menu.

config Use the *config* command to list the configuration files that are installed on the target. Configuration files are marked with an *m* in the first column if their size or checksum is different from the originally installed version. Use *config changed* to list the names of configuration files for which a corresponding *.O* (older) or *.N* (newer) version was created.

The *.O* version of a configuration file is the version of the file that was made obsolete by the software installation. The version without the *.O* suffix contains changes that are required for compatibility with the rest of the newly installed software, for increased functionality, or to correct bugs. Use *diff* or *gdiff* to compare the two versions of the files and transfer information that you recognize as machine- or site-specific from the *.O* version to the no-suffix version (see the *diff(1)* or *gdiff(1)* reference page).

When a *.N* version of a configuration file is created, it is a new, recommended version of the file. It contains changes or new features that can be added to the no-suffix version of the file at your option. Use *diff* or *gdiff* to compare the two versions of the files and add the *.N* version changes to the no-suffix version if you want them (see the *diff(1)* or *gdiff(1)* reference page).

date Use the *date* command with no arguments to display the system date. Enter arguments described in the *date(1)* reference page to change the date.

files Use the *files* command to list the names and other information about the files that make up a subsystem or group of subsystems. The other information includes any pending selection for the file (that is, *install*, *remove*, or *keep*), the installation state, the subsystem name, disk space delta, and file type. See “The files Command” on page 72 for more information.

<i>hardware</i>	Use the <i>hardware</i> command to display information about the internal architecture and components of the target system.
<i>help</i>	Use the <i>help</i> command to display Inst online help. Help is available on all topics listed in Table A-1.
<i>load</i>	Use the <i>load</i> command to open a file that was created with the <i>save</i> command to contain a custom selection of software. See also the discussion of <i>save</i> .
<i>mkfs</i>	<p>Use the <i>mkfs</i> command to create a new filesystem (replaces the <i>clean</i> command found in earlier versions of Inst). You can use <i>mkfs</i> in miniroot installations only.</p> <p>If a disk is new or if the current contents of the disk or filesystem are to be completely removed (cleaned), you can use <i>mkfs</i> to make new, empty filesystems. By default, new filesystems are made on the root (/) and user (/usr) filesystems. Partition 0 of a system disk is assumed to be the root filesystem (/); partition 6 (if present) is assumed to be the /usr filesystem.</p> <p>When you use <i>mkfs</i>, specify the name of the disk device to be used for the filesystem. For example, the following command</p> <pre>mkfs /dev/dsk/dks0d1s6</pre> <p>creates an empty filesystem on partition 6.</p> <p>All existing files, including all directories, programs, configuration files, and data, are lost when a new filesystem is made. You will be warned if a filesystem is already on the named device and prompted for a confirmation before any new filesystem is made.</p>
<i>mount</i>	Use the <i>mount</i> command without arguments to display mounted filesystems. Use <i>mount</i> with arguments to change the default mounts or add new mount points that are unknown to Inst. The two arguments are the block device name and the mount point. The <i>mount</i> command with arguments can be used during miniroot installations only.
<i>recalculate</i>	<p>Use the <i>recalculate</i> command to initiate a disk space calculation. The <i>recalculate</i> command identifies the current amount of free disk space, the size of each subsystem, and the amount of disk space that will be consumed if the subsystem is installed or freed if it is removed.</p> <p>If the <i>detailspacecheck</i> preference is off, or if files were created or removed in another shell, use <i>recalculate</i> to resynchronize space information.</p>

- relocate* Use the *relocate* command to store the installable image of a product in a non-default directory. To relocate a product, specify the product name (product images and subsystems cannot be specified) and the directory for storing the image (including a disk specification) as arguments to the command. Also use the *relocate* command without arguments to display a list of products that can be relocated.
- return* Use the *return* command to return control to the Main menu from the Administrative Commands menu. You can also use *return* to execute a Main menu command from the Administrative Commands menu, as shown in this example:
- ```
Admin> return go
```
- rqs* Use the *rqs* command to force the requickstart process on executable files and dynamic objects (see the *rqs(1)* reference page for details on *rqs* functions). This process is performed automatically during product installation, so this command is rarely required.
- save* Use the *save* command to copy the current custom selections of software to a file that you can later open using the *load* command (see also the discussion of *load*).
- set* Use the *set* command to examine and change the settings of variables called preferences. If you provide no arguments, the current settings are displayed. If you provide the name and a value, the preference value is changed.
- Most preference settings are saved in the installation history when you enter the *quit* command and are restored the next time that you invoke Inst. “The Preferences List” on page 128 describes *set* preferences in detail.
- sh* Use the *sh* command with no arguments to invoke an IRIX shell. Include a single IRIX command as an argument to execute the command immediately (see “The sh and shroot Commands” on page 75 for additional information).
- shroot* Use the *shroot* command with no arguments to invoke a “chrooted” IRIX shell (see the *chroot(1M)* reference page). If you provide arguments, they are taken as a single IRIX command to be executed immediately.
- The *shroot* command uses the *chroot(2)* system call to invoke the shell as if */root* were its root directory; that is, */root* becomes */* for all references. Thus files can be referred to with their normal pathnames (see “The sh and shroot Commands” on page 75 for additional information).

|                       |                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>space</i>          | Use the <i>space</i> command to display a disk space summary; the displayed summary is the same as the <i>list</i> command if the <i>set</i> preference <i>delayspacecheck</i> is off. If <i>delayspacecheck</i> is on, the <i>space</i> command forces the disk space calculation to be performed and displays the disk space summary. |
| <i>sethostname</i>    | Use the <i>sethostname</i> command to set the hostname of the target system; this command restarts networking using the new hostname. The <i>sethostname</i> command can be used in miniroot installations only.                                                                                                                        |
| <i>sethostipaddr</i>  | Use the <i>sethostipaddr</i> command to set the IP address of the target system; this command restarts networking using the new IP address. The <i>sethostipaddr</i> command can be used in miniroot installations only.                                                                                                                |
| <i>umount</i>         | Use the <i>umount</i> command to unmount disks that Inst has mounted by default or that have been mounted with the <i>mount</i> command. The filesystem name remains in the Inst device table even after the filesystem is unmounted. In some cases, filesystems mounted under the specified filesystem must be unmounted first.        |
| <i>updatekeepfile</i> | The <i>keepfile</i> file contains a list of new products (N designation in <i>list</i> output) in a distribution that are not designated for default installation. Use the <i>updatekeepfile</i> command to add to the <i>keepfile</i> file any new subsystems that are not currently selected.                                         |

## Interrupt Menu Commands

The list below describes all the commands that appear on the Interrupt menu.

|                 |                                                                                                                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>continue</i> | Use <i>continue</i> to resume processing from the point at which Inst was interrupted.                                                                                                                                                                                                                        |
| <i>help</i>     | Use the <i>help</i> command to display Inst online help. Help is available on all topics listed in Table A-1.                                                                                                                                                                                                 |
| <i>retry</i>    | Appears only if Inst displays the Interrupt menu automatically, which happens if an error occurs during <i>go</i> processing (subsystems are being installed or removed). Use <i>retry</i> after you correct the error (reported in a message above the menu) to repeat the installation and removal process. |

*set* [*preferences*] Use the *set* command to examine and change the settings of variables called preferences. If you provide no arguments, the current settings are displayed. If you provide the name and a value, the preference value is changed.

Most preference settings are saved in the installation history when you enter the *quit* command and are restored the next time that you invoke Inst. “The Preferences List” on page 128 describes *set* preferences in detail.

*sh* Use the *sh* command with no arguments to invoke an IRIX shell. Include a single IRIX command as an argument to execute the command immediately (see “The *sh* and *shroot* Commands” on page 75 for additional information).

*shroot* Use the *shroot* command with no arguments to invoke a “chrooted” IRIX shell (see the *chroot(1M)* reference page). If you provide arguments, they are taken as a single IRIX command to be executed immediately.

The *shroot* command uses the *chroot* system call to invoke the shell as if */root* were its root directory; that is, */root* becomes */* for all references. Thus files can be referred to with their normal pathnames (see “The *sh* and *shroot* Commands” on page 75 and the *chroot(2)* reference page for additional information).

*stop* Use the *stop* command to cancel the command in progress. If you use *stop* while Inst is reading the installation history or distribution inventory, or while it is calculating dependencies, Inst discards the information that it has read or calculated. Inst will then reread or recalculate this information before executing subsequent *list* commands, selection requests, or the *go* command.

If you use *stop* while Inst is installing software, it makes a record of pending selections. At the next *go* command, Inst restarts the installation process on any uncompleted product installations, then continues processing the selections that were pending.

**Caution:** Using the *stop* command during installation can leave software in an inconsistent state that requires a removal and reinstallation of the affected product.





## Troubleshooting Installations

This appendix presents information and procedures that you can use to troubleshoot problems that you might encounter during installation.

The appendix begins with a description of the types of error messages that Inst produces and then describes how to set Inst preferences to control the amount of feedback Inst provides. Most of the rest of the appendix discusses specific errors, arranged in groups according to how and when they are generated. Information and instructions are provided to help correct the error condition. A section at the end describes how to contact technical support and what information to have ready.

This appendix contains the following sections:

- “Types of Errors” on page 196
- “Resolving Errors” on page 197
- “Resolving Conflicts” on page 220
- “Resolving Network Problems” on page 225
- “Resolving Problems With CDs” on page 231
- “Checking Distribution Directories and CDs” on page 234
- “Using fx to Restore the Swap Partition” on page 235
- “Calling Your Support Organization” on page 236

## Types of Errors

Inst reports three types of errors: fatal, error, and warning. They are described in this section.

### Fatal

A fatal message occurs if Inst encounters a severe, unrecoverable error. An example of a fatal error is

```
FATAL: mmap of /dev/zero for 4000 pages failed
```

Inst exits immediately after encountering a fatal error. Fatal errors can be caused by hardware failures or lack of sufficient operating system resources, such as swap space. If a fatal error occurs, you might need to load the miniroot and take corrective action by modifying system configuration files.

### Error

An error message indicates that a command or other function was unable to complete successfully. For example

```
Inst> from /irix5.3
ERROR : The distribution /irix5.3 does not exist.
```

You may want to retry the operation after taking corrective measures.

### Warning

A warning indicates an unusual condition that Inst is prepared to handle but that may cause problems later during the installation. For example

```
% inst
WARNING : Starting up as Read Only (no installs or removals)
```

Inst will start but in read-only mode.

## Resolving Errors

This section discusses Inst error messages in detail. Error messages, possible reasons for the error, and possible solutions are provided. Errors are grouped according to how they are generated.

Error messages are shown in a typewriter-style font and are followed by indented explanatory text. For example

`Example of an error message`

This text describes the possible causes and solutions to the condition that produced the error.

Variables within the text of the error message (for example, hostnames) are shown in italics.

This section contains the following subsections:

- “Errors Loading the Miniroot” on page 197
- “Errors While Starting an Installation Session” on page 207
- “Preinstallation Check Errors” on page 212
- “Errors While Installing and Removing Software” on page 214
- “Errors Leaving an Installation Session (RQS Errors)” on page 220

### Errors Loading the Miniroot

This section discusses three types of errors that can occur when you are loading the miniroot. The following types are discussed:

- “Errors Loading the Miniroot From Local CD” on page 198
- “Errors Loading the Miniroot From Remote Host” on page 199
- “Other Errors Loading the Miniroot From a CD” on page 204

### Errors Loading the Miniroot From Local CD

```
dk<unit> error: unrecognized scsi disk drive
dksc(0,<unit>,8)sash.<cpu>: Command not found
```

If you see one of these errors while you are using a local CD during a miniroot installation, possible causes are

- The *dksc* command was not typed correctly: *--m* was entered instead of *-m*.
- There is no CD in the CD-ROM caddy.
- There is no caddy in the CD-ROM drive (for drives that use a caddy).
- The CD-ROM drive is not ready.
- A workstation with older PROMs that do not recognize CD-ROM drives is being used. An *fx*, *ide* (diagnostics), or *sash* was started and exited normally. Afterward, the older PROMs did not recognize the CD-ROM drive.
- A previous installation was terminated abnormally.

Perform the following actions to correct the error:

1. Check to make sure a caddy with a CD is in the drive.
2. Press the **Reset** button on the workstation main unit and begin the installation process again.
3. Follow the procedure in “Verifying That a CD-ROM Drive Is Recognized” on page 232.

```
CD-ROM drive not recognized
```

If you see this error while using local CD-ROM during an IRIX Installation, a possible cause is that IRIX does not recognize that the CD-ROM drive is present.

Perform these actions to correct the problem:

1. Shut down the workstation, verify that the CD-ROM drive is connected and turned on, and start the installation process over again.

2. If shutting down the workstation does not correct the problem, try turning the CD-ROM drive off and then on again.
3. See the section “Resolving Problems With CDs” on page 231.

```
dks0d3s8: Unexpected blank media: ASC=0x64
dks0d3s8: Can't read volume header
Error 20 while loading scsi(0)cdrom(3)partition(8)sashARCS
```

If you see these errors while using a CD distribution source during a miniroot installation, it is possible that the program *cdman(1)* was terminated, which left the CD-ROM drive in audio mode rather than in data mode.

To correct this problem, use the following procedure:

1. Quit Inst.
2. Return to the PROM Monitor.
3. Press the *Reset* button on the workstation.
4. Begin the installation again.
5. Refer to the section “Resolving Problems With CDs” on page 231 if problems persist.

### Errors Loading the Miniroot From Remote Host

In addition to the errors discussed in this section, refer to “Network Problem Diagnosis During Miniroot Installations” on page 229 for a discussion of network problems that may occur during a miniroot installation.

```
Cannot load network (0) bootp() server:path --could not connect to server
```

If you see this message, the cause is probably a conflict between the target’s IP address setting in the PROM and the address listed in the *hosts* file for the network. To correct this problem, use the following procedure:

1. Quit Inst.
2. Return to the Maintenance menu (see steps 1 through 4 of “The Software Installation Procedure” on page 38).
3. Start the command monitor from the System Maintenance menu. After your entry, the command monitor prompt (>>) is displayed.

4. Enter the *printenv* command to display the PROM setting for *netaddr*:

Find the value of the *netaddr* (network address) variable in the *printenv* output. If this value is different from the IP address that you obtained in “Getting the Target’s Name and Address” in Chapter 2, it must be reset.

5. Reset the *netaddr* variable to the value that you obtained from the *ping* command. Use this command to set the new PROM IP address:

```
>> setenv netaddr ping_IPaddress
```

```
No server for server:path(sashARCS)
Unable to load bootp()server:path(sashARCS):file not found
```

or

```
No server for server:CDir/dist(sashARCS)
open(bootp)server:CDir/dist(sashARCS) failed, errno = 6
Unable to load bootp()server:CDir/dist(sashARCS):file not found
```

If you see either of these errors during a miniroot installation, the cause might be an incorrect specification of the remote distribution source. To correct the error, enter the *setenv* command again. Specify the full, correct path to the distribution source and be sure to include the */sa* at the end of your specification. Then enter the *boot* command again.

```
TFTP error: I/O error (code 0)
Unable to load bootp()server:path>(sashARCS):
''bootp()server:path/sa(sashARCS)'' is not a valid file to boot.
```

or

```
TFTP error: Access violation (code 2)
bootp()server:path/sa(sashARCS): invalid
Unable to load bootp()server:path/sa(sashARCS):
''bootp()server:path/sa(sashARCS)'' is not a valid file to boot.
```

or

```
bootp()server:path/sa(sashARCS): invalid
Unable to load bootp()server:path/sa(sashARCS):
''bootp()server:path/sa(sashARCS)'' is not a valid file to boot.
```

If you see any of these errors after specifying a remote distribution during a miniroot installation, the problem might be one of the following:

- You have incorrectly specified *server*, *path*, or *cpu*.
- There is no physical connection between your workstation and the installation server.
- The *inetd.conf* file on the installation server has not been modified to allow *tftp*(1C) access to the CD-ROM mount-point directory or distribution directory.
- Routers between your workstation and the installation servers have not had *bootp*(1M) forwarding enabled (located in */etc/inetd.conf* on IRIX systems).
- A remote distribution directory is corrupted.
- You are trying to use a software distribution that does not include installation tools (the *sa* file).

To correct the problem, take one or more of these actions:

1. Check *server* (the installation server name), *path* (the distribution directory), and *cpu* (the CPU number) to make sure that you have spelled them correctly and that they exist.
2. Try to load the miniroot using the instructions in Chapter 3; you may see additional error messages that help you determine the cause of the problem.
3. Check the *inetd.conf* file on the installation server. The line containing *tftp* should be modified, and *inetd* should be restarted, as explained in “Setting Up an Installation Server” on page 16.
4. Check the *inetd.conf* file on each router between the target and installation server systems to verify that it has been modified, as explained in “Setting Up an Installation Server” on page 16.
5. Check Ethernet or other network cables and connections on the local and installation servers.
6. Check the *netaddr* variable on the target to make sure that it is set correctly (see Step 3 on page 226).

7. If possible, check the network connection to the installation server from a different system on the same network (see “Checking Network Connections” on page 225).
8. If the target system is a router (has multiple network connections), you might need to change its network connections so that the “normal” network device is connected to the same network as the installation server. Booting the miniroot is not supported over FDDI.
9. If necessary, bring up IRIX on the target system and check the network connection (see “Checking Network Connections” on page 225).
10. Check the distribution directory as described in “Checking Distribution Directories and CDs” on page 234.
11. Perform any additional procedures that are necessary to identify the problem. See the section “Resolving Network Problems” on page 225.

```
file file not found in server:path/sa; directory contains:
...
Unable to load bootp()...
```

or

```
File CDir/dist/sa not found on server server
Unable to load bootp(server:CDir/dist/sa(sashARCS)): no such file or
directory
```

If you see either of these errors during a miniroot installation, possible causes are

- The *boot* command contains a typing error in the *sashARCS*, *sashcpu*, or *sash64* portion of the command.
- The remote installation server name (*server*) was mistyped.
- The *netaddr* variable is not set correctly to your IP address (*IPaddress*).
- *bootp(1M)* is not running on the installation server.



Perform the following checks to debug this error:

1. Examine the last command that you entered and look for a spelling or capitalization error. Enter the command again with the correct spelling.
2. Check `/var/adm/SYSLOG` on the installation server to see whether it contains *bootp* messages. If *SYSLOG* contains *bootp* messages, *bootp* is running. The likely cause of the problem is that *netaddr* is set incorrectly on the target system.
3. If the installation server has multiple network interfaces, try specifying the hostname for each interface alternately. This sometimes resolves routing problems. To display the hostname for each interface, enter this command:
 

```
% /usr/etc/netstat -i
```
4. The `Address` column in the output contains hostnames.
5. Perform additional checks, as described in “Resolving Network Problems” on page 225.

```
Installation tools not found at server:CDdir/dist
```

or

```
Installation tools not found at server:path
```

If you see either of these errors during a miniroot installation from a remote distribution source, the CD or distribution directory that you specified might not contain installation tools. To correct this problem, confirm that the distribution source contains the installation tools (the *sa* file).

```
bootp()server:path/sa/(sashARCS) is not in a.out format
```

If you see this error message after you initiate miniroot loading, it may have one of the following causes:

- There are network problems or network traffic is too heavy to perform the installation.
- You specified an incorrect distribution directory name.
- The file *sa* in the distribution is damaged.

To resolve this problem, check the network using the procedures in “Resolving Network Problems” on page 225, or try again later.

### Other Errors Loading the Miniroot From a CD

```
Unable to load dksc(cntlr,unit,8)sashcpu: file not found
```

or

```
dksc(cntlr,unit,8)sashcpu: invalid
Unable to load dksc(cntlr,unit,8)sashARCS: file not found
```

or

```
open(bootp()server:CDDir/dist/sa(sashARCS)) failed, errno=2
Unable to load bootp()server:CDDir/dist/sa(sashARCS): file not found
```

If you see any of these errors during miniroot installation from CD, it may be that you are trying to load the miniroot from a CD that does not contain installation tools.

Switch to a CD that includes installation tools to load the miniroot, then switch back to your original CD.

### Questionable Miniroot Image: IRIX 6.3 and Earlier

**Note:** The following error message occurs only during installations of IRIX 6.3 or earlier. If you are installing IRIX 6.4 or later, refer to “Questionable Miniroot Image: IRIX 6.4” on page 205.

```
root and swap are on the same partition. Either the system is
misconfigured or a previous installation failed. If you think the miniroot
is still valid, you may continue booting using the current miniroot image.
If you are unsure about the current state of the miniroot, you can reload
a new miniroot image. Finally, you may abort the installation and return
to the PROM; in this case you will need to use the 'fx' program to correct
the disk label information. See the 'Software Installation Guide' chapter
on Troubleshooting for more information.
```

```
Enter 'c' to continue booting the currently loaded miniroot.
Enter 'r' to reload the miniroot.
Enter 'a' to abort the installation.
Enter your selection and press ENTER (c, r, or a)
```

This error occurs on personal workstation (such as O2 and Indy systems) when you try to load the miniroot after a power failure or if a system restart is performed during the installation. If you attempted a system restart instead of loading the miniroot after the interruption occurred, you were automatically placed in the version of Inst that is part of the previously installed miniroot.

Take one of the following actions to correct the condition:

1. Enter **c** if you want to install software with the currently loaded miniroot. Entering **c** and quitting Inst fixes the boot information. You can then restart the system after Inst is loaded without using the *fx* utility.
2. Enter **r** if you want to reload the miniroot. You might want to do this if, for example, the current version of the miniroot is corrupt or if you want to load another version of the miniroot.
3. Enter **a** to abort the installation and to go back to the command monitor. You can do this, for example, if you are running IRIX 6.2 or earlier and you want to use the *fx* command. Also use this selection on IRIX 6.3 systems if you want to abort the installation and restart the system.

**Note:** If you are running IRIX 6.2 or earlier and want to use *fx* to correct boot information and boot from the root partition, refer to the procedure documented in “Using *fx* to Restore the Swap Partition” on page 235.

#### **Questionable Miniroot Image: IRIX 6.4**

**Note:** The following error messages occur only during installations of IRIX 6.4 or later. If the target is running IRIX 6.3 or earlier and you see a similar error, refer to “Questionable Miniroot Image: IRIX 6.3 and Earlier” on page 204.

```
It appears that a miniroot install failed. Either the system is
misconfigured or a previous installation failed.
If you think the miniroot is still valid, you may continue booting
using the current miniroot image. If you are unsure about the
current state of the miniroot, you can reload a new miniroot image.
```

```
You may abort the installation and return to the menu, or you can
fix (reset to normal) the miniroot install state.
See the 'Software Installation Guide' chapter on Troubleshooting for
more information.
```

Enter 'c' to continue booting the old miniroot with no state fixup.  
Enter 'f' to fix miniroot install state, and try again.  
Enter 'r' to reload the miniroot.  
Enter 'a' to abort (cancel) the installation.

Enter your selection and press ENTER (c, f, r, or a)

This error is posted when you try to reboot the miniroot after a power failure or a system reset interrupted the miniroot load process, risking the integrity of the miniroot.

You can safely choose **c** if you were using the miniroot normally when the interruption occurred or if you unintentionally exited the miniroot. Do not use **c** if the interruption occurred while the miniroot copy was in progress and the process did not conclude with a posting of the `Inst>` prompt. In this case, assume that the miniroot image is compromised and select either **f** to fix, **r** to reload the miniroot, or **a** to abort the installation. The **f** selection is also appropriate if you deliberately reset the system while using the miniroot and you want the system to boot normally.

The **r** selection performs the entire miniroot load process from the beginning, removing any effects of the previous load attempt. If you need to preserve the load state to demonstrate the error condition to others (such as your support organization), chose **a** to abort the installation. Unlike **f** or **r**, the **a** choice makes no changes in the system state; it simply returns control to the PROM monitor.

It appears that a miniroot install failed. Either the system is misconfigured or a previous installation failed. You may continue booting without fixing the state. You may abort the installation and return to the menu, or you can fix (reset to normal) the miniroot install state. See the 'Software Installation Guide' chapter on Troubleshooting for more information.

Enter 'c' to continue with no state fixup.  
Enter 'f' to fix miniroot install state, and try again.  
Enter 'a' to abort and return to menu.

Enter your selection and press ENTER (c, f, or a)

This error is posted if you perform a system reset after a power failure or a previous system reset interrupted the miniroot load process, risking the integrity of the miniroot.

You can safely choose **c** if you were using the miniroot normally when the interruption occurred or if you unintentionally exited the miniroot. Do not use **c** if the interruption occurred while the miniroot copy was in progress and the process did not conclude with a posting of the `Inst>` prompt. In this case, choose **f** to fix the miniroot image or **a** to abort the copy and return to the `Inst` menu.

## Errors While Starting an Installation Session

The error discussions in this section are grouped as follows:

- "Wrong Diskless Modes" on page 208
- "Errors Starting Live Installations" on page 209
- "Inst Library libinst.so Errors" on page 210
- "Errors in the Distribution" on page 210

In general, check what `Inst` is using as the default distribution. You may have to use the *from* command to point it at the desired distribution.

### Wrong Diskless Modes

```
ERROR : Unable to start inst: The target appears to be a diskless client
tree, since the file /root/var/inst/.client is present. Restart in
client- mode using client_inst (1M). If you are certain that you want to
run in non-diskless mode, remove the client file /root/var/inst/.client
and restart inst.
```

```
FATAL ERROR : Unable to start inst: The target appears to be a share tree
for diskless client, since the file /var/inst/.share is present. Restart
in share-mode using share_inst(1m). If you are certain that you want to
run in non-diskless mode, remove the file /var/inst/.share and restart
inst.
```

These messages mean that Inst believes that the target is a diskless client tree because in a previous installation Inst was invoked in “diskless” mode reserved for the diskless installation tools *share\_inst* and *client\_inst* (see the *share\_inst(1M)* and *client\_inst(1M)* reference pages).

If the target has been previously created as a diskless tree, then continuing with a normal (nondiskless) installation would severely corrupt the installed software. You should only attempt diskless installations using *share\_inst* and *client\_inst* (see the *share\_inst(1M)* and *client\_inst(1M)* reference pages).

However, if you are certain that the target is not used for diskless installations, remove the files */var/inst/.share* and */var/inst/.client* (or, if in the miniroot, */root/var/inst/.share* and */root/var/inst/.client*). Then restart Inst.

If you are performing a miniroot installation, Inst will exit abnormally and prompt you to restart the system (*y*), enter Inst (*n*), or start a shell (*sh*). Choose *sh*:

```
Ready to restart the system? (y, n, sh) sh
rm /root/var/inst/.share
rm /root/var/inst/.client
exit
```

(You use only the */root* prefix to the path for miniroot installations.) Then return to Inst:

```
Ready to restart the system? (y, n, sh) n
...
Inst>
```

### Errors Starting Live Installations

These errors occur when starting Inst from IRIX.

Sorry! The system is not set up for non-miniroot installations of all the selected subsystems, since the configuration file `/var/inst/inst_special` is missing. Try the installation again from the miniroot.

You may not perform a live installation of some subsystems (labeled with *b* by the *list* command) without the *inst\_special* configuration file present. If you are unable to obtain this file from another system, you must perform the installation from the miniroot.

Another inst is currently running

You may not have two copies of Inst running in read/write mode to the same target simultaneously. The second session is run in read-only mode.

Inst determines this by looking for a file called `$rbase/var/inst/inst.lock`. (*\$rbase* is the root directory for the current software installation.) In rare cases, it may be necessary to remove this file by hand.

A previous installation session was not completed successfully.

This error means a previous version of Inst was interrupted or killed before it completed all the actions requested by the user. Information on the state of the last session has been saved in the file `$rbase/var/inst/checkpoint`. For more information on recovering from the checkpoint file, see "If Inst Is Interrupted" on page 57.

### Inst Library *libinst.so* Errors

The Inst products—*inst*, *swmgr*, *showfiles* and *showprods*—all link with the *libinst.so* dynamic object. If, when starting one of these programs, an *rld* error appears regarding *libinst.so*, it is probable that you have an incompatibility between the binary and *libinst.so*. In this situation, it is best to reinstall *coe.sw.base* from the miniroot to get the latest versions of these products.

```
26379:inst: rld: Fatal Error: cannot map soname 'libinst.so' using any of
the filenames
/usr/lib/libinst.so:/lib/libinst.so:/lib/cmplrs/cc/libinst.so:/usr/lib/
cmplrs/cc/libinst.so:
-- either the file does not exist or the file is not mappable (with reason
indicated in previous msg)
```

This error message means that the *libinst.so* file is missing.

```
852:swmgr: rld: Error: unresolvable symbol in swmgr:
post__15VkDialogManagerFPccPFP10_WidgetRecPvT2_vN22PvT1P14VkSimpleWindo
w
```

This error message indicates that the *libinst.so* file is present but not the right version.

### Errors in the Distribution

```
ERROR : No such host: host
```

This error can appear after executing a command that requires access to a distribution through the network.

The most likely cause is a bad hostname. Check the hostname and use the *from* command to set the correct distribution location.

If the hostname appears correct and there was a delay before the error message appeared, it is possible that your system is experiencing network problems. See the section “Resolving Network Problems” on page 225.



ERROR : The distribution *dist:/pathname* does not exist.

This error occurs when a command attempts to reference the distribution but the distribution path references a nonexistent directory or a product file. For example

```
Inst> from dist:/sgi/baddir
Connecting to dist ...
ERROR : The distribution dist:/sgi/baddir does not exist.

Inst> from dist:/sgi/hacks/badprod
Connecting to dist ...
ERROR : The distribution dist:/sgi/hacks/badprod does not exist.

Inst> from /host/dist/sgi/baddir
ERROR : The distribution /host/dist/sgi/baddir does not exist.
```

Determine the correct pathname and use the *from* command to set the correct distribution location.

ERROR : The product *host:/path/sc* is bad.

This error occurs if the distribution specified references a file that is not a valid product file. For example

```
Inst> from /usr/tmp/file
ERROR : The product /usr/tmp/file is bad.
```

Note that when referencing an individual product, the product file must be used. In the following error, the product was incorrectly specified using the *idb* file:

```
Inst> from dist:/sgi/hacks/sc.idb
Connecting to dist ...
ERROR : The product dist:/sgi/hacks/sc.idb is bad.
```

The product *sc* should be specified as follows:

```
Inst> from dist:/sgi/hacks/sc
Connecting to dist ...
```

ERROR : The distribution *host:/path* does not contain any products.

This error results when the distribution directory specified does not contain any product files. You must specify the correct distribution directory.

*Missing products in listing*

If a product *prod* appears in a distribution directory along with its *idb* file (*prod.idb*) and image files (*prod.image ...*), but does not appear in the product listing in Inst, then the product files may be corrupt.

Use *ls* to make sure that the product files are in the distribution directory. Make certain that you are viewing all the products in the distribution by executing the following commands:

```
Inst> view dist
Current Location : distribution
Inst> view filter all
Inst> list
```

If the product is still not visible, the product was not read in and the product files are probably corrupt. See the section “Checking Distribution Directories and CDs” on page 234 for more information.

## Preinstallation Check Errors

When you enter the *go* command, Inst executes the preinstallation check before installing any files. If any errors are detected during this check, Inst lists the problems and returns to the main menu *without installing or removing software*.

```
Not enough space on / for the new unix kernel
Not enough space on /usr for requickstart overhead (see rqs(1))
Not enough space on /usr for the installation overhead
Not enough space on / (additional 85kbytes required)
```

These errors mean that you need to make more disk space available by removing files (in these examples on the */* and */usr* filesystems), or select fewer subsystems for installation.

A live installation usually requires extra temporary disk space. Because some of the files to be upgraded are currently in use, either by the operating system or by running applications, Inst must maintain multiple copies of these files during the installation and, in some cases, it must maintain the files until you reboot the computer.

If you are performing a live installation, there might be enough available disk space for all the new software but not enough additional temporary disk space to complete the installation. In this situation, try closing some applications and then entering the *go* command again. If there is still not enough space, you may have to run the installation in the miniroot.

You can save the selections that you have already made by creating a selections file; when you finish removing software, load the file to reapply your selections. Use this sequence of commands during either a live or a miniroot installation:

```
Inst> admin save filename
Inst> keep *
Inst> remove subsystem subsystem ...
Inst> admin load filename
```

The installation request will install or remove files in the following nfs-mounted filesystems:

*/filesystem*

Please cancel or confirm the request.

1. Cancel the installation request
2. Continue the installation request

Please enter a choice [1]:

Inst issues these warnings to protect against accidental installation of files into NFS-mounted directories. Normally software installations are made on the local host. If you really want to install across an NFS mount, choose 2; otherwise cancel the installation (1), return to the Main Menu, and use the *keep* command to install fewer subsystems.

**Note:** To disable this confirmation, set the preference *confirm\_nfs\_installs* to *off*.

ERROR: Subsystems cannot be installed/removed because they contain files on read-only NFS-mounted filesystems. You may de-select these subsystems for install/remove; or, if you wish to install these subsystems locally, you must first unmount these filesystems, or remove any symbolic links pointing into them (check the pathnames of the Files listed below); or, if you wish to install these subsystems onto the remote filesystems, you must unmount them and then re-mount them as read-write.

```
Filesystem: /usr/share NFS-mounted, read-only

Subsystem File
insight.sw.data /usr/share/Insight/lib/addBooklist
vino.man.pages /usr/share/catman/a_man/cat7/vino.z
```

Installations and removals canceled

ERROR: Subsystems cannot be installed/removed because they contain files in write-protected directories. You may de-select these subsystems for install/remove; or, if you wish to continue with the current selections, you must first change the directory permissions using the `chmod(1M)` command.

```
Subsystem File
ViewKit_dev.sw.demo /usr/share/src/lib/ViewKit/Utilities/list.c++
dwb.sw.dwb /usr/share/lib/tmac/tmac.e
```

Installations and removals canceled

Any of these messages mean that you lack the appropriate permission to install all the files in the selected products.

This is usually an indication that you are using NFS to share filesystems on a remote host and that some of the subsystems selected for installation install files into those remote filesystems.

Check your selections to make sure you are not installing or removing “shared” software, such as online books or reference pages. Use the *keep* command to deselect those products.

## Errors While Installing and Removing Software

This section contains the following subsections:

- “Disk Space Errors” on page 215
- “Subcommand and Exitop Errors” on page 217
- “Network Time-out Errors” on page 218
- “Archive Corrupt Errors” on page 219
- “Device Busy Errors” on page 219

These errors cause the Error/Interrupt menu to appear automatically (see Figure B-1).

## Error/Interrupt Menu

- |                      |                                               |
|----------------------|-----------------------------------------------|
| 1. stop              | Terminate current command                     |
| 2. continue          | Continue current command                      |
| 3. set [preferences] | List all preferences or set/clear a preferenc |
| 4. help [topic]      | Get help in general or on a specific word     |
| 5. sh [cmd]          | Escape to a shell or run a command            |
| 6. shroot [cmd]      | Escape to a chrooted shell or run a command   |

Interrupt&gt;

**Figure B-1** The Interrupt Menu

If the preinstallation check completes without errors, Inst begins installing and removing files. If an error occurs after this point, Inst stops and presents the interrupt menu. First try to correct the cause of the error and then choose *retry* from the interrupt menu.

If this does not work, or you are unable to correct the problem, you can choose *stop* to cancel the installation immediately and return to the Main menu.

If you *stop* the installation, the current image in progress (such as *oe.sw*) will be in an inconsistent state (partially installed/removed). The installation history will not have been updated for these subsystems (*oe.sw.\**). You are strongly advised to either reinstall these products (select *go* at the main menu to restart the installation from the beginning of the partial image) or, for products not marked “required,” remove them completely.

**Disk Space Errors**

Despite efforts to accurately predict the required disk space, Inst may occasionally fail during the installation with an error such as this:

```
ERROR : An error occurred while Installing new versions of selected product
subsystems
Write of pathname failed: No space left on device
```

This produces the Error/Interrupt menu (see above). Use the *shroot* command to enter the shell. Remove or compress unnecessary large files, exit the shell, and retry the operation. If you are unable to locate any expendable files, stop the installation and choose fewer subsystems for installation.

The sequence that follows illustrates how to resolve the previous error by removing and compressing files:

```

Interrupt> shroot
df
Filesystem Type blocks use avail %use Mounted on
/dev/root efs 1939714 1939702 12 100% /
ls -l /usr/tmp/core.*
-rw----- 1 guest guest 20971520 Oct 20 01:00 /usr/tmp/core.0
-rw----- 1 guest guest 0 Oct 20 01:00 /usr/tmp/core.1
-rw----- 1 guest guest 3145728 Oct 20 01:01 /usr/tmp/core.3
rm /usr/tmp/core.0 /usr/tmp/core.1
compress /usr/tmp/core.3
df
Filesystem Type blocks use avail %use Mounted on
/dev/root efs 1939714 1892566 47148 98% /
exit

Interrupt> retry
Installing new versions of selected pv.man subsystems
Installing new versions of selected pv.sw subsystems

```

If there is still not enough disk space, consider the possibility that you may not need some large files on your workstation. The list below gives filenames relative to root, but remember that, if you are doing a miniroot installation, */root* must be prepended to each of the filenames if you escape to the shell with *sh*. If you escape to the shell with *shroot* or are performing a live installation, use the filenames as given. Look for these large files:

- Kernel core dump files in */var/adm/crash/vmcore\** and */var/adm/crash/unix\**.
- Files put into */lost+found* and */usr/lost+found* by *fsck(1M)*.
- If you have process accounting enabled, the directories *fiscal*, *nite*, and *sum* in */var/adm/acct* may contain large files.
- */var/adm/SYSLOG*. If this file is very large, you might want to truncate or remove it. Because it is in use, you must notify the daemon. Use these commands:

```

rm /var/adm/SYSLOG
killall -HUP syslogd

```

- Unnecessary files in */tmp* and */usr/tmp*. Note that */usr/tmp/inst.a\** files are temporary files created by *inst* and should not be removed.
- Core files. You can find all core files in the root and user filesystems with these commands if you are performing a miniroot installation:

```

Admin> shroot
/bin/find / /usr -mount -name core -print

```

- Large user files that can be temporarily archived to tape. One way to find large files is to use the `sysadm(1)` command if it is installed (`sysadm` is not included in Release 5.0 and later):

```
% /bin/su -
Password:
/usr/bin/sysadm filesize
```

Running subcommand 'filesize' from menu 'filemgmt', FILE MANAGEMENT

Enter full pathname of the directory to search [?, q]: `/usr/people/joe`

Enter the number of large files to be included in list (default 10 largest) [q]: `10`

The largest 10 files in `/usr/people/joe`:  
(report)

You can also use the `find` command to find files that are larger than, for example, 2,000 blocks (1 MB):

```
find / -local -size +2000 -print
```

- If your site uses NFS, you may be able to NFS-mount reference pages installed on another workstation rather than installing them on your workstation when disk space is scarce.
- Replacing your system disk with a larger disk is sometimes an option when you need more disk space in order to install the software you want. NFS mounting remote filesystems is also an option when you need more disk space because Inst installs software onto NFS-mounted filesystems if those filesystems are exported read write from the remote host and if your permissions (usually those of the superuser) allow write access to the necessary directories on that host.

### Subcommand and Exitop Errors

As part of the installation procedure, Inst executes *subcommands*. These are UNIX shell commands that perform initialization functions specific to each product. For example, some products use subcommands to install a custom icon in the system Icon Catalog. Some subcommands, called exit commands, or *exitops*, run at the end of the installation and sometimes originate from more than one subsystem.

```
Stderr: Cannot create pathname: No such file or directory
ERROR : An error occurred while Installing new versions of selected product
subsystems
Command "command"
```

If a subcommand fails during the installation of a specific product, an interrupt menu is also presented. The subcommands that run at the end of the installation, during the “exit-commands” phase, may affect multiple subsystems. Inst displays any errors from these exitops but does not present the Interrupt menu.

If an Interrupt menu is presented, try to gauge from the error message the cause and severity of the problem. The error could indicate that the affected product will not function completely or correctly or that the system might fail to boot. Decide whether to ignore the error and continue, to fix the problem and retry, or to stop and return to the Inst Main menu.

Consult the release notes of any affected product for further information. For example, the release notes may specify a particular order in which the software subsystems must be installed in order to function properly.

### Network Time-out Errors

```
Connecting to host ...
host.domain: Interrupted system call
Host host is not responding, retrying
host.domain: Interrupted system call
Host host is not responding, retrying
host.domain: Interrupted system call
ERROR : Timed-out waiting for host
```

Inst presents the Error/Interrupt menu. See the section “Resolving Network Problems” on page 225 to determine the cause of the network failure. You may need to continue the installation at a later time, depending on the availability of that host.

If the network is merely slow, or the server is heavily loaded, use the *set* command to raise the value of the *timeout* and/or *network\_retry* preferences.



### Archive Corrupt Errors

```
File filename not in compressed format
Compressed input file is corrupt (internal overflow)
Unexpected EOF
Can't open archive: archive
Archive archive is in an unrecognized format
Archive archive is corrupt
```

Inst is unable to properly extract files from the software distribution, which is compressed in a special format. If you are installing over a network, check the system logs for signs of network errors (see “Resolving Network Problems” on page 225).

If you are performing a live installation, you may need to use a newer version of the installation tools; when the distribution format is upgraded, older versions of Inst cannot always read more recent software distributions (new versions of Inst can read older distributions formats, however). Use Inst from the miniroot, preferably the miniroot that accompanies the software upgrade you are trying to install.

### Device Busy Errors

```
filesystem: Device Busy
```

There may be a file open in the named filesystem if you get this error. Quit Inst and then reinvoke it to force it to close the open file.

For example, if you were trying to unmount all filesystems from Inst Admin:

```
Admin> umount -a
</root/usr: Device Busy error messages>
Admin> return
Inst> quit
Ready to restart the system. Restart? { (y)es, (n)o, (sh)ell, (h)elp }
n
Inst> admin
Admin> umount -a
```

## Errors Leaving an Installation Session (RQS Errors)

An error has occurred while requickstarting your system. No loss of functionality occurred.

A requickstart failure simply indicates that some files were not requickstarted. The net effect is that the startup time of the failed binary will be slightly slower than had it been successfully requickstarted. The error message will also provide the name of a log file where there is a detailed explanation of the RQS error(s). See *rqs(1)* for a detailed explanation of requickstart.

```
Sproc of /usr/etc/rqsread failed
```

```
Sproc of /usr/etc/rqsall failed
```

```
/usr/etc/rqsread terminated abnormally
```

```
/usr/etc/rqsall terminated abnormally
```

These messages indicate that you probably need to upgrade your system to get newer versions of these files.

```
/usr/etc/rqsread terminated abnormally due to signal #
```

```
/usr/etc/rqsall terminated abnormally due to signal #
```

These messages indicate that the named process was killed by a signal. The relevant signal number will be provided so it will be possible to determine the cause of the termination.

## Resolving Conflicts

Installation conflicts occur when there are unsatisfied product dependencies or when incompatible product are selected for installation. If Inst detects conflicts when the user enters the *go* command, the conflicts must be resolved before the installation is carried out. See “Step 6: Resolving Conflicts” on page 52 for more information on the procedures used to resolve conflicts.

The error you see when a conflict has been detected is

```
Inst> go
ERROR : Conflicts must be resolved.
(conflict description and options)
```

This section is divided into the following subsections:

- “Resolving Different Types of Conflicts” on page 221
- “Overriding Conflicts” on page 224

**Note:** The 10-digit number that appears in the conflict message is the product’s version number. You can use the `showprods -n` command to display product version numbers.

## Resolving Different Types of Conflicts

The different types of conflicts discussed here are:

- “Preexisting Conflicts” on page 221
- “Required Product Conflict” on page 222
- “Prerequisite Conflict” on page 223
- “Back Prerequisite Conflict” on page 223
- “Incompatible Products Conflict” on page 223
- “Downgrade Conflict” on page 224

### Preexisting Conflicts

Unresolved product dependencies and incompatible installed products may already exist on the target before the current Inst session. These conflicts are known as preexisting conflicts. For example, the following conflict shows an installed product that depends on another product that is not installed.

```
- Existing Conflict -
swmgr.sw.eoe is installed but is missing prerequisites:
 1a. Also remove swmgr.sw.eoe (1021391900)
 1b. Install eoe.sw.unix (1010852020 - 2147483647)
```

Normally, these conflicts are not visible. If they are, you have three options.

1. Resolve the conflicts by using the procedures in “Step 6: Resolving Conflicts” on page 52.
2. Choose to ignore preexisting conflicts by setting the resource *show\_existing\_conflicts* to *false* and marking something for removal or installation to force a recalculation of the installation rules. Because *show\_existing\_conflicts* is a permanent resource, this ignores pre-existing conflicts for future installations as well.  

```
Inst> set show_existing_conflicts false
Inst> remove prod1
Inst> go
```
3. Choose to ignore conflicts for this installation session. Refer to “Overriding Conflicts” on page 224.

### Required Product Conflict

Required product conflicts occur when a required subsystem is not marked for installation or when a required subsystem is marked for removal.

```
subsystem is required and must be installed
 1a. Also install subsystem (xxxxxxxxxxx)
```

This type of conflict occurs when the distribution contains a required subsystem that is not installed and not marked for installation. In this case, the only solution is to install the required subsystem.

```
subsystem is required and may not be removed - sorry!
 1a. Do not remove subsystem (xxxxxxxxxxx)
```

This type of conflict occurs if you mark a required subsystem for removal. In this case, the only solution is to retain the required subsystem.

### Prerequisite Conflict

This type of conflict occurs when a product is marked for installation and it requires a product that is not already installed or marked for installation.

*product* cannot be installed because of missing prerequisites:

- 1a. Do not install *product* (xxxxxxxxxx)
- 1b. Install *product* (xxxxxxxxxx - xxxxxxxxxxx)

To resolve the above conflict, you have two options. The first is to not install the product whose requirements are missing. The second is to install the required products. In the above example, the required product is not on the current distribution. You must locate a distribution that contains the required product and install it before continuing with this installation. See “Step 2: Specifying the Source” on page 41 for further instructions.

### Back Prerequisite Conflict

This type of conflict occurs when a product is selected for removal but other products depend on it.

*product* cannot be removed because other products depend on it.

- 2a. Do not remove *product* (xxxxxxxxxx)
- 2b. Also remove  
*product1* (xxxxxxxxxx)  
*product2* (xxxxxxxxxx)

There are two options for resolving this type of conflict. You can choose not to remove the product or remove all the products that depend on it.

### Incompatible Products Conflict

This type of conflict occurs when a product is marked for installation and is incompatible with another product that is marked for installation or with one that is already installed.

*product* (xxxxxxxxxx) is incompatible with *product1* (xxxxxxxxxx)

- 2a. Do not install *product* (xxxxxxxxxx)
- 2b. Do not install *product1* (xxxxxxxxxx)

This conflict is the result of selecting two incompatible products for installation. Install only one.

- product* (xxxxxxxxxx) is incompatible with *product1* (xxxxxxxxxx)
- 2a. Do not install *product* (xxxxxxxxxx)
  - 2b. Also remove *product1* (xxxxxxxxxx)

This conflict is the result of marking a product for installation that is incompatible with a product that is already installed on the target. To resolve this type of conflict, choose one of the two incompatible products to be on the target.

### Downgrade Conflict

This type of conflict occurs when a product that is marked for installation is an older version of a product that is already installed.

- You have marked *product.old* (xxxxxxxxxx), which is an older version of *product.new* (xxxxxxxxxx)
- 1a. Replace *product.new* (xxxxxxxxxx) with *product.old* (xxxxxxxxxx)
  - 1b. Do not install *product.old* (xxxxxxxxxx)
  - 1c. Set resource *neweroverride* to value true

The first option replaces the newer, installed version of the product with an older one. The second option retains the newer version of the product on the target. The last sets the resource *neweroverride*, which allows the installation of older products for newer ones without conflict. Unless there is a valid reason, the newer product should be retained on the target.

### Overriding Conflicts

If installation conflicts cannot be resolved but the installation must be done, you can override the conflicts and continue with the installation by setting the *rulesoverride* preference. Overriding conflicts leaves a preexisting conflict on the target, which can be seen if the *show\_existing\_conflicts* preference is set to *true*. This action is recommended only for extreme cases and for knowledgeable users because it can introduce unfulfilled product dependencies or install incompatible products on the system.

**Note:** Resolve all conflicts possible before using the *rulesoverride* feature.

In the following example, a conflict is “resolved” by setting *rulesoverride* to *true*:

```
product1 cannot be installed because of missing prerequisites:
 1a. Do not install product1 (xxxxxxxxxx)
 1b. Install product0 (xxxxxxxxxx - xxxxxxxxxxx) (not on current distribution)

Inst> set rulesoverride true
Inst> go
Inst> set rulesoverride false
```

## Resolving Network Problems

This section discusses how to check network connections from IRIX and how to diagnose network problems during an Inst session.

### Checking Network Connections

The steps below explain several tests and checks that you can perform from IRIX to verify that your workstation is connected to an installation server. (Note that if you are in Inst, you can access IRIX with the *sh* command.)

1. Test the connection for Inst user access:

```
% /usr/bsd/rsh server -l user date
```

In the previous command, *server* is the name of the installation server and *user* is the user ID you are using for installation. Normally, *user* is “guest.” If the date is not returned, you have specified the wrong *server*; there is a network problem, or *user* is not a valid user ID. (See “Configuring an Installation Account” on page 18 for more information about *user*.)

2. Test the TCP/IP connection:

Connections to installation servers are done over Terminal Control Protocol/Internet Protocol (TCP/IP) in a manner similar to *rsh* (see the *rsh(1C)* reference page). A simple test of this connection can be done by using *ping* (see the *ping(1M)* reference page):

```
% /usr/etc/ping -q -f -s 2048 -c 100 server
PING <server> (<IPaddress>): 2048 data bytes
----<server> PING Statistics----
100 packets transmitted, 100 packets received, 0% packet loss
round-trip (ms) min/avg/max = 0/2/7
%
```

In the previous example, *server* is the name of the installation server. If you see packet loss, you could have a problem with your network connection. If you receive a message that the host is unknown, verify the name of the host you are trying to contact and be sure you are typing it correctly.

This network connection test is not possible if you are performing a miniroot installation; if you are, test the connection before beginning the installation, if possible.

3. Check the setting of the *netaddr* NVRAM variable.

In some situations, you might have network problems if the IP address of your workstation in its nonvolatile random-access memory (NVRAM) does not match its IP address in */etc/hosts*. A mismatch can occur when you move a workstation, but it does not cause a problem until you attempt to load the miniroot for a software installation. You can check the IP address in the NVRAM on your workstation while you are using IRIX by giving this command:

```
% /etc/nvram netaddr
```

From the Command Monitor, you can check the IP address in the NVRAM with this command:

```
>> printenv netaddr
```

If the four-part number returned from either command does not match the IP address in */etc/hosts* on your workstation, you may be able to change it from IRIX (not all models of workstations support changing NVRAM from IRIX):

```
/etc/nvram netaddr localIPaddress
```

Or change it from the Command Monitor:

```
>> setenv netaddr localIPaddress
```



4. Verify that the installation server allows *tftpd* access (required for miniroot installation only).

“Enabling TFTP Access on an Installation Server” on page 17 describes the procedure for verifying that the installation server has been modified to allow *tftp* access (see the *tftpd(1M)* reference page).

To get more debugging information, add the *-l* argument to the *tftp* line in */etc/inetd.conf* and restart *inetd* (see the *inetd(1M)* reference page). The line should look like this:

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -l
```

Debugging information is written to */var/adm/SYSLOG*.

5. Verify that routers between your workstation and the installation server forward *bootp* packets (see the *bootp(1M)* reference page).

“Enabling BOOTP Forwarding on Routers” on page 16 describes the procedure for verifying that routers have been modified to allow *bootp* access.

To get more debugging information, add the *-d* argument to the *bootp* line in */etc/inetd.conf* and restart *inetd* (see the *inetd(1M)* reference page). The line should look like this:

```
bootp dgram udp wait root /usr/etc/bootp bootp -f -d
```

Debugging information is written to */var/adm/SYSLOG*.

For more information on networking, see the *IRIX Admin: Networking and Mail* and the *NFS Administration Guide*.

### Network Problem Diagnosis During an Inst Session

If the network is slow (usually indicating network problems), Inst may appear to be frozen for long periods (much greater than the time-out time); in reality it may be reading a few bytes at a time, timing out, retrying, then reading a few more bytes. This sort of behavior, as well as any error messages regarding network time-outs or retries, is an indication that it may be desirable to investigate the condition of the network.

Here are some of the common error messages that might occur during a remote installation session:

-- Host *'host'* is not responding, retrying

The remote host did not respond in a reasonable amount of time; we will retry a few times before giving up. See the preference “network\_retry” for a discussion of how to control the number of retries Inst makes before it gives up.

-- Timed-out waiting for *'host'*

The remote host has timed out several times in a row. A serious network problem probably needs to be resolved before we can continue. Either resolve the network problem and continue the installation or cancel the install and try again later.

On a slow network, changing the network time-out (by setting the preference “timeout” to the new time-out in seconds) may be necessary, though in general this is not recommended and will probably not cure a real network problem but will only increase the length of time before such a problem is reported to the user.

-- Can't set up network connection to host *host*: *reason*

You were unable to establish an initial network connection to the remote host for the reason given. See “Checking Network Connections” on page 225 for a discussion of what to do to test the network.

-- Lost connection to *host*

The network connection was broken. This probably means that the remote host is down.

-- No such host: *host*

The host is not listed in the host table. This could mean that the hostname was mistyped, that there is something wrong with the */etc/hosts* file, or that the NIS or DNS server is down.

-- Couldn't parse ls output from remote host: `'ls_output'`

Running an `ls` command on the installation server succeeded but gave output different from what was expected. The only immediate solution is to install a different `ls` program (for example, a POSIX compatible `ls` program) on the installation server.

-- Failed reading remote directory `dir: error_msg`

It was not possible to perform an `ls` command on the installation server. The given error string should give some clue as to the likely reasons:

- **Illegal option:** The installation server has an `ls` command that takes a different set of arguments from the ones the local `ls` command takes. The only immediate solution is to install a different `ls` program (for example, a POSIX-compatible `ls` program) on the installation server.
- **Cannot access ... No such file or directory:** The remote directory does not exist
- **Cannot access ... Permission denied:** The given user does not have sufficient permissions to access the directory.

-- Can't get shell/tcp network port to host `host: error_msg`

The service “exec” or “shell” was not found on the system. This probably means that there is something wrong with the file `/etc/services` or the services map on the NIS server.

### Network Problem Diagnosis During Miniroot Installations

These messages come from the boot PROM rather than Inst.

-- unable to load bootp() `machine: /path/sa(sash.IPXX)`

-- bootp() `machine: /path/sa(sash.IPXX)` is not a valid file to boot

To find out if the file exists, enter the following command on the installation server:

```
mkboottape -f /pathname/sa -l |grep sash
```

In the previous example, `pathname` is the distribution directory; for example, `sa` may be in `dist:/irix/6.3`.

Chances are that the file does exist and that this is probably a network problem—trying to *bootp* through too many gateways. Because the *bootp* protocol is not as robust as it could be, it is sometimes unable to find a file on a server if network traffic is heavy. By installing from an NFS-mounted file system on a local network, NFS is dealing with the gateways and *bootp* is only going across the local network.

This can also occur when the server is a multihomed machine.

```
-- no server for machine:/path/sa(sash.IPXX)
```

```
-- unable to load bootp()machine:/path/sa(sash.IPXX): file not found
```

This could mean that the *netaddr* variable is set incorrectly. From the command monitor, enter a *printenv netaddr* command and see if the address returned is set to an address on the local subnet (see step 3 of “Checking Network Connections” on page 225 for more information).

```
-- Error 7 while loading network(0)bootp()mach1:/path/sa(sashARCS)
```

```
-- UX:csh: ERROR: ../swindow - Command not found
```

The previous message might appear when loading from an automounted distribution (for example, *machine:/hosts/...*).

These errors indicate that you need to modify */etc/inetd.conf* (*/usr/etc/inetd.conf* on pre-5.x systems) on the installation server (*mach1* in this example) and remove the “-s /usr/local/boot /usr/etc/boot” from the *tftpd* entry (or alternatively, add the desired pathname to the end of the list of accessible paths after -s). Then restart *inetd*:

```
/etc/killall -HUP inetd
```

```
-- panic free'ing mbuf while loading miniroot
```

This is a problem in the IP20 prom.

Check your Ethernet cable—make sure that it is connected tightly to your machine. If you still see this problem, you can try to load the miniroot from a local machine via an NFS mount. This problem rarely occurs, and only when there is high network traffic.

-- Cannot load network(0)bootp()*machine: /path*

-- Problem reading file magic id, err 0 cnt0

This can mean network problems, such as a bad router between the current machine and the installation server.

-- Unable to load bootp()*machine: /path: ``bootp()*machine: /path*''* is not a valid file to boot

You are trying to boot from a file that is not a valid *sa* image.

If you are booting from the command monitor, be sure to specify the *sa* file, not just the distribution directory containing the *sa* file.

-- No remote connection

This is probably an internal error resulting from trying to access a remote connection that used to be open but is now closed.

-- Unable to locate your password information (user-id *UID*)

Your current user ID is not a valid account on the system. This unlikely error probably means there is something wrong with the host table (*/etc/hosts*) or with the NIS server.

-- Can't open network connection: no hostname!

No remote hostname was given. You must supply the name of the network host.

## Resolving Problems With CDs

### Checking CD-ROM Drives

To check CD-ROM drives, you must verify that the system recognizes the drive and that the CD you want to use is mounted. These procedures are described in the following sections.

## Verifying That a CD-ROM Drive Is Recognized

The procedure to verify that a CD-ROM drive is recognized depends on your situation:

- If IRIX is running, enter the *hinv* command:

```
% hinv
```

For each CD-ROM drive, you should see one line of output. For example,

```
CDROM: unit 4 on SCSI controller 0
```

If you do not see a line of output for a CD-ROM drive, it is not recognized.

- If you are in the miniroot, escape to a shell with the *shroot* command and enter the *hinv* command:

```
hinv
```

For each CD-ROM drive, you should see one line of output. For example,

```
CDROM: unit 4 on SCSI controller 0
```

If you do not see a line of output for a CD-ROM drive, it is not recognized.

- If you are in the Command Monitor, enter the *hinv* command:

```
>> hinv
```

For each CD-ROM drive, you should see one line of output. Some examples:

```
SCSI CDROM: dksc(0,4)
```

```
SCSI CDROM: scsi(0)cdrom(4)
```

```
SCSI Disk: dksc(0,4)
```

These examples show the CD-ROM drive on an older workstation. The CD-ROM drive is recognized, but it is shown as a disk. If you do not see a line of output for a CD-ROM drive, it is not recognized.

When a CD-ROM drive is not recognized, it is usually because the CD-ROM drive was not powered up properly. If it is an external drive, the CD-ROM drive must be powered on before the workstation main unit is powered on.

The procedure for making the system recognize the CD-ROM drive depends on whether you are running IRIX or the miniroot:

1. If you are running IRIX, exit `Inst` if it is running, warn other users, shut the workstation down with `shutdown` (see the `shutdown(1M)` reference page), or use System Shutdown on the System menu, then reboot the workstation to bring up IRIX again.
2. If you are in the miniroot, get back to the PROM Monitor, press the *Reset* button on the workstation main unit and then bring up `Inst` again. If this does not fix the problem, turn the CD-ROM drive off and then on again.

### Verifying That a CD Is Mounted

When using a CD-ROM drive, the CD that contains the software you want to install must be mounted. Mounting is done automatically by `Inst` when using a local CD-ROM. The user executing `Inst` must have root privileges in order for it to be able to mount the CD. When using a remote CD-ROM, the mounting must be done separately on the remote server. To verify that the CD is mounted, use the `df` command, below (see the `df(1)` reference page). If you are using a local CD-ROM, escape to a shell. If you are using a remote CD-ROM, enter the command on the installation server.

For example,

```
% /bin/df
Filesystem Type blocks use avail %use Mounted on
/dev/root efs 1939714 1749520 190194 90% /
/dev/dsk/dks0d4s7 efs 828672 817805 10867 99% /CDROM
```

Look at the directory name on the right. For a local CD-ROM, you should see `/CDROM`. For a remote CD-ROM, the name `/CDROM` is likely but another directory name for the mount point (called *CDdir* in this document) may have been chosen.

If the CD is mounted, list the files it contains to verify that you have the correct CD inserted.

## Checking Distribution Directories and CDs

To verify that a distribution directory or a mounted CD contains the right files, the workstation that contains the distribution must be running IRIX. Change directories to the distribution directory (*distdir* or *CDdir/dist*) and list the files with *ls*. Files in software distributions have these names:

```
mr
sa
product
product.idb
product.images
```

The file *sa* is used for miniroot installations only, so it does not need to be present if you are doing an IRIX installation. The file *mr* may or may not be present; there is no problem if it is missing. More than one product file, *product*, may be in the directory. Each product requires a file called *product.idb* (installation database) and one or more *product.image* files. Examples of product files are *ee* and *maint\_nfs*. Common values of *images* are *man* and *sw*.

If a distribution does not have the correct files, the most likely causes are that the distribution directory was not copied correctly or that the files in the distribution directory were modified after it was created. Use *distcp -c* to compare the original with the copy (see the *distcp(1M)* reference page) and, if there is a discrepancy, copy the original distribution again.

Copy the distribution directory using *cp -r* (or *rcp -r* for a remote copy) and then use *distcp -c* to compare the original with the copy (see the *distcp(1M)* reference page).

If you suspect that the contents of the files in the distribution directory have been corrupted, try installing from the CD that was used to create the distribution directory.

An additional check is possible for CDs: change directories to *CDdir* and list the files with *ls*. One file and at least two directories should be listed:

```
RELEASE.info
dist
relnotes
```

*RELEASE.info* is an ASCII text file that contains release information. *dist* contains the product files, and *relnotes* contains the release notes, which describe the exact contents of each CD. Use *relnotes* to read the release notes (see the *relnotes(1)* reference page).



## Using fx to Restore the Swap Partition

The following procedure corrects the condition described in “Questionable Miniroot Image: IRIX 6.3 and Earlier” on page 204, in which you receive the error message `root` and `swap` are on the same partition. Use this procedure on systems running IRIX 6.2 or earlier IRIX versions. The procedure modifies the boot information to cause the system to boot from partition 0 (the root partition) instead of partition 1 (the swap partition) where the miniroot currently resides. (This discussion assumes that you are using the 4D1-4.0 or later version of *fx*.)

**Caution:** The *fx* command is intended for advanced users and should not be used unless you have a definite need for it. Refer to the *fx*(1) reference page for complete information on using *fx*.

1. Enter the command monitor.

Follow steps 1 through 4 of “Loading From a Local CD” on page 30 to display the System Maintenance menu, then select `Enter Command Monitor`.

2. Invoke *fx* from the standalone version on your workstation or from a local or remote CD-ROM. See the *fx*(1M) reference page for complete instructions on invoking *fx* or use the following procedure.

- To invoke the standalone copy of *fx* on your workstation, enter

```
>> boot stand/fx --x
```

- To invoke *fx* from a CD with installation tools in a CD-ROM drive on your workstation, enter this command:

```
>> boot -f dksc(cntlr, unit, 8) sashARCS dksc(cntlr, unit, 7) stand/fx.ARCS --x
```

In the previous command, *cntlr* and *unit* are the controller and unit numbers of the CD-ROM drive (see “Getting CD-ROM Device Numbers” on page 23).

- To invoke *fx* from a CD with installation tools mounted on an installation server named *server*, enter this command:

```
>> boot -f bootp() server:CDdir/stand/fx.ARCS --x
```

In the previous command, *CDdir* is the mount point directory for the CD (for example, */CDROM*).

3. Run *fx* to restore the boot file to the root partition. The sample session below shows the *fx* defaults for system disk device name, controller number, and drive number. Unless your system configuration is not standard, the *fx* defaults are correct and you can respond with <Enter> to the prompts.

```
Obtaining /stand/fx from server server
n+n+m entry: p
fx version 4.0 IP22, Aug 23, 1991
fx "device-name" = (dksc) <Enter>
fx: ctrlr# = (0) <Enter>
fx: drive# = (1) <Enter>
...opening dksc(0,1,)
...controller test...OK
Scsi drive type == CDC 94171-9 0184

----- please choose one (? for help, .. to quit this
 menu)-----

[exi]t [d]ebug/ [l]abel/ [a]uto
[b]adblock/ [ex]ercise/ [r]epartition/ [f]ormat
fx> label/create/boot
```

At this point the System Maintenance menu appears, and you can restart your system or restart the installation.

## Calling Your Support Organization

Silicon Graphics, Inc., provides a comprehensive product support maintenance program for its products.

If you are in the United States or Canada and would like support for your Silicon Graphics supported products, contact the Technical Assistance Center at 1-800-800-4SGI. If you are outside these areas, contact the Silicon Graphics subsidiary or authorized distributor in your country.

If you have read the troubleshooting information in this chapter and still need help, have this information available when you call your support organization:

- the serial number of your workstation (required)
- the products that you are trying to install and their release numbers (see the CD label)

- the release numbers of software products that are currently installed (use *showprods*, described in the *showprods(1M)* reference page)
- the type of software distribution you are using (local or remote CD-ROM or distribution directory)
- the text of any error messages you have seen
- the hardware configurations of your workstation and any installation server used for installation (model numbers, the size of your system disk, and so on)



---

# Index

## A

- abbreviating commands, 85-90
- admin* command, 180
- administration, system
  - documentation, xxiii-xxiv
- Administrative Commands menu
  - commands (illustration), 71
  - functions, 70-76
  - hidden commands, 120
  - reference, 180
- archive corrupt error, 219
- automatic
  - selection of subsystems, 181
- automatic initialization, 11, 42
- automatic installation, 85, 147
- automatic miniroot installation, 147-165
- automatic selections
  - during initialization, 42
  - for hardware changes, 103
  - in Inst sessions, 50
  - in noninteractive installations, 85
  - in software upgrades, 96
  - on empty systems, 7
  - turning off, 125
- autoselect* option of *set*, 181
- autoselect preference, of *set* command, 125
- auxiliary commands, 183

## B

- backing up targets, 22-23
- base release, 3
- BOOTP, 16, 152
- bootp* command, 227
- bootp* errors, 200-203
- bootp forwarding
  - default setting, 16
  - enabling, 16-17
- boot* PROM Monitor command, 202
- boot server, 29, 147, 151, 152

## C

- CD-ROM drive
  - controller and unit number, 20
  - creating a directory for, 21
  - device numbers, 23
  - stopping daemon, 21
- CDs
  - ejecting, 20, 21
  - mounting, 21
  - organizing for installation, 23
- checkpoint restart
  - file, 57
  - handling failures, 58
  - session recovery, 57
- chroot* command, 191
- clean* command (obsolete), 190

- close* command, 180
  - command reference, 177
  - commands
    - abbreviating names, 86
    - entering by number, 85
    - hidden, 120
    - keywords in entries, 87
    - product and image names in entries, 87
    - reference, 177
    - shortcuts, 85-90
    - View menu, 64
    - wildcards, 86
    - wildcards in entries, 86
  - compact discs *See* CDs.
  - compatibility errors, avoiding, 108
  - config* command, 189
  - configuration directory, 148, 153
  - configuration files
    - backing up, 22
    - creating *.N* versions, 60
    - creating *.O* versions, 60
    - effects of *mkfs* command, 76
    - listing, 61
    - managing, 60-62
    - merging, 61-62
    - notice of changes, 61
    - not removed, 109
    - removing, 62
    - retrieving from tape, 22
  - configuration server, 147, 152
  - conflict command
    - using, 53
  - conflicts
    - error messages, 53
    - in patches, 101
    - resolving, 52-53, 220-225
  - conflicts command
    - overview, 13
    - reference, 180
  - continue command
    - reference, 192
  - continue* command, to resume a session, 81
  - corrupt archive error, 219
  - CPU specification error, 201
- D**
- date and time, resetting, 33, 35, 72
  - default subsystems
    - defined, 7
    - in noninteractive installations, 85
    - on empty target systems, 7
  - delayspacecheck* option of *set*, 181, 192
  - delayspacecheck* preference, of *set* command, 122
  - deltas* command
    - reference, 187
  - delta size, 64, 68
  - detailspacecheck* preference, of *set* command, 74, 122
  - device missing messages (during miniroot loading), 33, 35
  - device names
    - CDs, 235
  - device numbers
    - in CD mount command, 21
  - diagnostic procedures
    - checking network connections, 226
  - diskless client error, 208
  - disk partitions, 29, 150, 154, 190
  - disk space
    - controlling calculations, 121-123
    - delaying calculations, 122
    - delta information, 189
    - detailed space checks, 121
    - disabling calculations, 122
    - displaying summaries, 74
    - estimated space checks, 121

- disk space (*continued*)
    - forcing recalculations, 74
    - freeing, 75
    - minimizing calculation time, 123
    - not enough, 7, 179
    - recalcule* command, 190
    - requirements for individual files, 73
    - requirements in *list* output, 46
  - disk space error, 215
  - displaying software information, 43-45
  - dist command
    - reference, 187
  - dist* preference, of set command, 124
  - distribution, defined, 3
  - distribution directories
    - contents of, 20
    - creating, 20-22
  - distribution directory
    - contents of, 21
    - copying to, 21
    - defined, 8
    - limiting *tftp* access, 18
    - structure, 21
  - distribution inventory
    - displaying, 43
    - sample display, 44
    - viewing, 12, 65
  - distribution keyword, 90
  - distribution path error, 199
  - distribution server, 147
  - distribution source
    - changing, 178
    - default, 12, 181
    - defined, 8
    - displaying available, 41
    - displaying current, 83, 124
    - \$dist* variable, 83
    - opening additional, 42
    - selecting, 15
    - specifying, 41-42, 83-84
    - specifying for a miniroot load, 34
    - specifying one product, 84
    - specifying with a user account, 84
    - types, 8-9
    - types of (illustration), 9
  - dk* error, 198
  - dksc* error, 198
- ## E
- ejecting CDs, 20, 21
  - error
    - abnormal session interruption, 57
    - archive corrupt, 219
    - conflicting selections, 52-53
    - CPU specification, 201, 202
    - diskless client, 208
    - distribution, 58, 210-212
    - exit operations (*exitop*), 217
    - fatal, 196
    - installation processing, 54
    - IP address, 202
    - leaving installation, 54-56, 220
    - libinst.so*, 210
    - live installation, 209
    - loading miniroot, 197-207
    - local CD, 198
    - network timeout, 218
    - not enough space, 212, 215-217
    - path, 199
    - pre-installation, 212
    - read-only directories, 54, 213
    - remote host, 199
    - rld, 210
    - RQS, 220
    - selection, 59
    - server, 199
    - server specification, 202

error (*continued*)  
  session terminated abruptly, 57-59  
  software compatibility, 108  
  subcommand, 217  
  types, 196  
  warning message, 196  
escaping to the shell, 191, 193  
*/etc/hosts* file, 226  
*/etc/inetd.conf* file modifications, 16-18  
exiting Inst, 183  
exit operation (exitop) error, 217

## F

factory installed software, 8  
fatal error, 196  
feature release, 4, 95-99  
files  
  displaying, 72-73, 123-124  
  identifying types, 73  
  in a subsystem, 189  
  target inventory, 73  
files command  
  reference, 189  
*files* command, 72-73  
filesystems, 192  
  creating, 76  
  for miniroot installations, 33, 35, 75  
  mounting, 76  
  unmounting, 76  
filesystems command  
  reference, 187  
filter (*view* command argument), 66  
filter command  
  functions, 66  
  reference, 187

from command  
  formats, 41  
  overview, 12  
  purpose, 41  
  reference, 180  
  specifying target software, 42  
*fx* command, 235

## G

*go* command  
  effects of, 51  
  overview, 12  
  processing, 51-52  
  reference, 181  
*guest* account. *See* user account.

## H

*hardware* command, 190  
hardware-specific files  
  change notice, 103  
  in software product releases, 4  
  updating, 103-104  
*help* command  
  how to use, 14  
  reference, Administration menu, 190  
  reference, Interrupt menu, 192  
  reference, Main menu, 181  
  reference, View menu, 188  
hidden commands, 120  
*hinv* command  
  device information, 20, 23  
hostname, obtaining, 24



## I

- idb file, 5, 118
- idb* files, 5
- images
  - defined, 5
  - naming convention, 5
  - specifying in commands, 87
  - specifying with keywords, 87
  - viewing level, 67
- .index* file, 153
- index* file, 160
- inetd.conf* file modifications, 201
- initialization script, 112-114
- initialization. *See* automatic initialization.
- Inst
  - abnormal interruption, 57
  - defined, 3
  - help feature, 14
  - invoking, 10, 84-85
  - invoking in live installations, 36
  - invoking in miniroot installations, 34, 36
  - quitting, 55
- installation
  - checks, 181
  - history database, 188, 191, 193
  - state of files, 189
- installation database
  - and automatic initialization, 42
  - defined, 5
  - effects of *mkfs* command, 76
- installation methods, 10, 27
- installation order, planning, 23-25
- installation procedure
  - cursor instructions, 177-179
  - detailed description, 38-57
  - illustrated summary, 38-39
- installation processing, 51-52
- installation server, 147
  - access limitation from the miniroot, 75
  - defined, 8
  - obtaining hostname and address, 24
  - security, 18
  - selecting, 16
  - setting up, 16-22
- installation server error, 199
- installation user account. *See* user account.
- install command
  - arguments, 50
  - function, 48
  - reference, 183
  - sample entries, 49
- inst command
  - for automatic installations, 85
  - for live installations, 36
  - for specifying distribution source, 84
- inst.init* file, 112-114
- Inst Main Menu, 191
- Inst overview
  - from *help* command, 14
  - operational description, 10-13
- inst* utility, 3
- interactive mode, 184
- Interrupt menu, 79-81, 214
- inventory. *See* distribution inventory *and* target inventory.
- invoking Inst, 84-85
  - for live installations, 36
- IP address
  - for making filesystems, 76
  - obtaining, 24
- IP address in NVRAM and IRIX, 226
- IRIX administration
  - documentation, xxiii-xxiv
- IRIX commands, in the miniroot, 75
- IRIX version information, 33, 35

**K**

keep command  
  arguments, 50  
  function, 48  
  reference, 183  
  sample entries, 49  
keyword arguments, 90  
keywords, 87-90

**L**

legend. *See list command.*, 44  
*level* command, 67, 188  
*libinst.so* error, 210  
licensing, 169  
list command  
  column contents in output, 45  
  controlling with *view*, 64-69  
  default output, 43  
  displaying the distribution inventory, 43  
  interpreting output, 45-46  
  legend, 43, 44, 141  
  mnemonic tags, 46  
  overview, 12  
  patch items in output, 100  
  reference, 183  
listing software information. *See displaying software information and the list command.*  
live installation  
  defined, 10  
  disk space requirements, 36  
  invoking *Inst*, 36  
live installation error, 209  
*load* command, 190  
loading the miniroot  
  from a remote source, 34-36  
  from local CD, 30-34  
  informational messages, 35

**M**

Main menu  
  features, 11-13  
  hidden commands, 120  
  illustration, 11  
Main menu (illustration), 40  
maintenance release, 4, 95-99  
man pages. *See reference pages.*  
mediad daemon  
  restarting, 22  
  stopping, 21  
menus  
  controlling, 120-121  
  hidden commands, displaying, 120  
  *See also individual menu names.*  
  using commands on inactive menus, 63  
miniroot  
  defined, 10  
  IRIX command availability, 75  
miniroot configuration file, 154-160  
miniroot installation  
  automatic, 147-165  
  conditions requiring, 101, 103, 109  
  defined, 10  
  effects on *sh* and *shroot* commands, 75  
  local load procedure, 30-34  
  remote load procedure, 34-36  
miniroot installation file, 148  
mkfs command  
  purpose, 77  
  reference, 190  
  side effects, 76  
mnemonic tags, in *list* output, 46-101  
*more?* prompt, 181  
*mount* command, 190  
mounting CDs, 21  
mounting filesystems, 76  
mount point directories, 190, 192

*mrconfig* file, 153

*mrconfig* file. *See* miniroot configuration file.

## N

*netaddr* PROM variable, 200, 201, 226

network

checking connections, 225

preparations, 16-19

problem resolution, 225-231

timeout error, 218

network address

displaying IP setting, 24

displaying PROM setting, 200

network timeout, 125

*neweroverride* preference, of *set* command, 105

noninteractive installation, 85

*nvr* command, 226

## O

older releases, installing, 105-108

*open* command, 42, 183

opening multiple distributions, 23

optional software, installing, 99

overlays, 3

*overprint* preference, of *set* command, 124

## P

partitions, disk, 190

password protection, 19

patches

conflicts, 101

defined, 4

installing, 99-102

removing, 102

patch release

identifying in *list* output, 100

path specification error, 199

permanent preferences, 126

*ping* command, 16, 226

preferences

displaying current settings, 119

file (*.swmgrrc*), 126

for menu control, 120-121

overview, 126

permanent, 126

tables, 128-143

transient, 126

preliminary selections, 7, 11, 42, 50

product

*See* software products.

product description file, 5

product descriptions

reading, 11, 42, 180

PROM

displaying variables settings, 200

network address setting, 200

## Q

quit command

overview, 13

reference, 183

quitting a session, 55-56

## R

*\$rbase* root installation directory, 126

recalculate command

reference, 190

*recalculate* command, 74

reference pages, installing, 104-105

reinstalling software, 103-104  
release family, 4  
release notes  
  how to read, 3  
relocate command  
  reference, 191  
remove command  
  arguments, 50  
  function, 48  
  reference, 183  
  sample entries, 49  
removing  
  all software, 109  
  files, 183  
required subsystems, defined, 7  
restarting workstation, 183  
resuming an interrupted session, 81  
*retry* command, 192  
return command  
  reference, Administration menu, 191  
  reference, View menu, 188  
.rhosts file  
  modifying for installation, 19  
rld error, 210  
RoboInst server, 147  
routers  
  identifying, 16  
  use in miniroot installations, 16  
rqs command  
  reference, 191  
*rsh* command, 225  
*rulesoverride* preference of the *set* command, 109

## S

*sa* file in software distributions, 203  
*sash* program, 29, 148, 150, 152

*save* command, 191  
security  
  on Inst user account, 19  
  *tftp*, 17  
selecting software (for installation), 48-51, 90-93  
selection  
  automatic after *from*, 181  
  current, 181  
  initial, 181  
  of subsystems, 183  
  pending, 57  
selection commands  
  *install*, *remove* and *keep*, 48-50  
  overview, 12  
  *step*, 90-93  
selection processing, 51-52  
selections file, 113  
server. *See* installation server.  
server specification error, 199  
*set* command  
  reference, Administrative menu, 191  
  reference, Interrupt menu, 193  
  reference, Main menu, 184  
  reference, View menu, 188  
  saved option values, 188, 191, 193  
*sethostipaddr* command, 192  
*sethostname* command, 192  
*sh* command  
  reference, Administration menu, 191  
  reference, Interrupt menu, 193  
  reference, Main menu, 184  
*sh* command, to invoke an IRIX shell, 75  
shell (argument to *set* command), 75  
shell, IRIX, 75, 191, 193  
*SHELL* environment variable, 75  
shortname  
  defined, 4  
*show\_files* preference, of *set* command, 123

- show* command, 188
- showfiles command
  - for changed configuration files, 61
  - to list configuration files, 109
- showfiles* command, 22
- showprods command
  - for displaying release numbers, 108
  - for new systems, 8
  - for reference pages, 105
- showprods* command, 237
- shroot command
  - reference, Interrupt menu, 193
  - reference, Main menu, 184
- shroot* command, 191
- shroot* command, to invoke an IRIX shell, 75
- shutdown* command, 30
- size of subsystems
  - in *list* and *step* displays, 46
  - in *list* and *step* displays, 64
  - with delayed space checking, 122
- sizes command
  - reference, 188
- software
  - old versions, 183
  - removing, 183
- software inventory. *See* distribution inventory and target inventory.
- Software Manager, 3, 37, 126
- software options, installing, 99
- software product release, 3
- software products
  - components, 5-8
  - defined, 4
  - naming convention, 4
  - patching, 99-102
  - reinstalling after hardware changes, 103-104
  - reinstalling files after patching, 102
  - removing, 109
  - removing files after patching, 102
  - specifying in commands, 87
  - specifying with keywords, 87
  - viewing level, 67
- sort* command, 68, 188
- space checking
  - delaying, 122
  - detailed, 121-122
  - estimated, 121
  - overview, 121
- space command
  - reference, 192
- space* command, 74
- spec file, 5, 118
- step command
  - controlling with *view*, 64-69
  - reference, 184
- step* command, 90-93
- stop command
  - reference, 193
- stopping an installation, 80
- subsystems
  - default, 7
  - defined, 7
  - displaying files in, 72
  - files in, 189
  - installing incompatible, 108
  - naming convention, 7
  - required, 7
  - sizes. *See* size of subsystems.
  - specifying in commands, 86
  - specifying with keywords, 87
  - viewing level, 67
- support, 236
- swap partition, in miniroot installations, 29
- .swmgrrc* file, for preferences, 126
- sysadm* command, 217

system administration  
  documentation, xxiii-xxiv  
System Maintenance menu  
  accessing, 31  
  illustration, 31

## T

tardist file, 116  
*targ* command, 189  
target  
  defined, 8  
  IP address, 24  
  shutting down, 30  
target inventory  
  displaying, 44  
  displaying on new systems, 8  
  removing, 109  
  sample display, 45  
  specifying as the source, 42  
  viewing, 12, 65  
  viewing files in, 73  
TFTP, 17, 152, 153  
tftp  
  enabling access, 17-18, 227  
*tftp*, enabling access, 201  
*tftpd* daemon, 17  
*tftp* security, 17  
*timeout* preference, of *set* command, 125  
transient preferences, 126

## U

*umount* command, 192  
unmounting filesystems, 76

updatekeepfile command  
  reference, 192  
user account  
  configuring for Inst access, 18-19  
  non-*guest*, 18, 19  
  specifying in *from* commands, 84

## V

*/var/inst/patchbase* file  
  function, 100  
  reinstalling contents, 102  
  removing contents, 102  
view  
  filtering with keywords, 66  
  setting level, 67  
  setting sort order, 68  
  setting to distribution or target, 65-66  
view command  
  overview, 12  
  reference, 186  
View Commands menu  
  functions, 64-69  
  hidden commands, 120  
  illustration, 64

## W

warning, 196  
wildcards, in commands, 86



---

## Tell Us About This Manual

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

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

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

Please send the title and part number of the document with your comments. The part number for this document is 007-1364-110.

Thank you!

## Three Ways to Reach Us

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

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