Software Installation Administrator’s Guide

Document Number 007-1364-050

CONTRIBUTORS

Written by Pam Sogard, with contributions by Susan Ellis and John Raithel
Illustrated by Derrald Vogt
Edited by Christina Cary
Production by Lorrie Williams
Engineering contributions by Ken Chin, Sunita Das, David Fenstemaker,
Paul Jackson, Mark Minnihan, Akkana Peck, Will Rusch, and Jeffrey Thomas
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson,
Erik Lindholm, and Kay Maitz

Software Installation Administrator’s Guide
Document Number 007-1364-050

Contents

Examples

Figures

Tables

About This Guide

The *Software Installation Administrator's Guide* explains how to install Silicon Graphics® software products using Inst, the command line interface to the installation utility, *inst*(1M). This guide 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. The *Software Installation Administrator's Guide* provides instructions for performing conventional installations and specialized installations that rely on system firmware (miniroot installations).

When to Use This Guide

As of IRIX™ 5.3, the *inst* utility includes Software Manager, a graphical user interface to installation. Software Manager is documented separately in the *Personal System Administration Guide*.

Because of its graphical features, Software Manager cannot be used for miniroot installations. To perform miniroot installations, use Inst and the instructions provided in Chapter 3 of this guide. Information in Chapter 1 explains how to determine whether a miniroot installation is necessary.

Use Chapter 8 as a comprehensive reference to managing the preferences that appear in the Preferences window of Software Manager. Chapter 8 describes the available preferences and explains how to manage the configuration files for preferences.

The *Software Installation Administrator's Guide* 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 *Software Installation Administrator's Guide* is organized into nine chapters and an appendix, which provide this information:

Chapter 1, "Introduction"
: 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 that should be done on a server and target system to ensure smooth installations.

Chapter 3, "Starting a Miniroot Installation"
: Describes how miniroot installations work and how to prepare for them. Contains procedures for starting a miniroot installation from different distribution media on various models of Silicon Graphics systems. Use this chapter in conjunction with the procedures in Chapter 5 to install software that requires a shutdown of fundamental IRIX services.

Chapter 4, "Starting a Live Installation"
: Explains how to start an installation on a running system. Use this chapter in conjunction with the procedures in Chapter 5 to perform live software installations.

Chapter 5, "Using Inst"
: Contains procedures for using Inst to perform a software installation during either a miniroot installation (Chapter 3) or a live installation (Chapter 4).

Chapter 6, "Inst for Experts"
: Describes optional methods for using Inst that supplement the basic method presented in Chapter 5. In general, procedures in this chapter require mastery of Chapter 5 material.

Chapter 7, "Maintenance Tips"
Contains procedures for updating installed software and keeping software current with site requirements.

Chapter 8, "Tailoring Inst Operation"
Explains how to customize Inst to suite both personal preferences and the requirements of other system users. Includes information for tailoring Software Manager.

Chapter 9, "Inst Command Reference"
Describes all commands that are available from Inst menus, including hidden commands. Commands are ordered alphabetically by menu.

Appendix A, "Installation Troubleshooting"
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 *Software Installation Administrator's Guide* is experienced UNIX® system and site administrators who are responsible for installing software on various models of Silicon Graphics workstations and servers. This guide 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 non-standard filesystem configurations, proficiency with UNIX filesystem 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, filenames, directory names, commands, and command arguments 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
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.

- *IRIX Advanced Site and Server Administration Guide* for information on IRIX filesystem management and on system and 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.

Part I
Routine Installation Procedures

Chapter 1

Introduction

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*(1M), is to transfer distribution software, which has an encoded format, to a hard disk in a format that is usable. 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).

The media containing installable software that you purchase from Silicon Graphics is referred to as a *distribution*. Most distributions are not specific to a particular model of Silicon Graphics computer; distributions ordinarily contain all versions of any hardware-specific files that might be required. Sometimes, however, a new server or personal workstation model is introduced, accompanied by a special software distribution. When this occurs, subsequent distributions are fully compatible with the new model.

Software Product Releases

A *software product release* contains all software required to support a given version of a product and the tools that are needed to install the release. One or more software product releases are contained in a distribution (see Figure 1-1).

When you install a software product release, files from previous versions of the release are automatically removed before the new files are installed (the exception is configuration files, which are saved if they contain local modifications). If a product release includes hardware-specific files, the installation utility automatically determines the file version that is needed on a particular model and installs that version.

Product releases may have prerequisites that require a particular installation order. They might also have compatibility requirements with other installed products. Inst protects users against potential problems by preventing installation if there are any unresolved incompatibilities or prerequisites (unless you override this safeguard).

Installation instructions, including prerequisites and incompatibilities information, are provided in the product release notes, which are always included as an installable subsystem of the product. 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.

Figure 1-1 illustrates the hierarchical structure of a software distribution that comprises three software product releases.

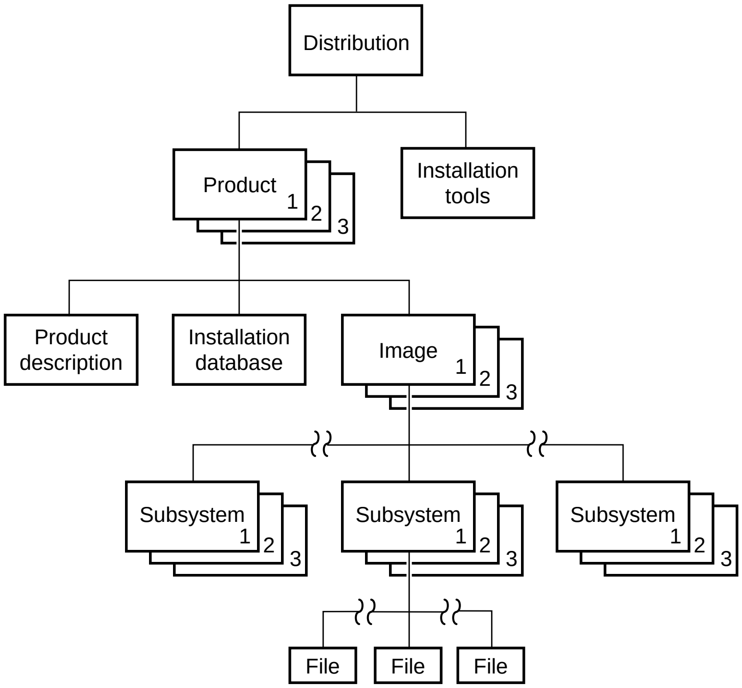


Figure 1-1 Components of a Distribution

Anatomy of a Software Product

A *software product* is a collection of files that support either a germane 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.

A software product comprises three elements: a *product description*, an *installation database*, and one or more *images*, which are composed of *subsystems*. 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 *ftn_dev*.

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*.

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*.

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.

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 "Patch Releases" on page 7 for exceptions). Subsystems are assigned three-part names of the form product.image.subsystem. 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. Where 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).

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; on targets with software installed, the default selections are all upgrades to the installed software that the distribution contains. When subsystems are displayed during an installation session, default subsystems are tagged with the letter *d* and required subsystems are tagged with the letter *r*.

Factory Installed Subsystems

Some subsystems from the *eoe1*, *eoe2*, *motif_eoe*, and other standard (no-cost) products are factory installed. The *eoe1* and *eoe2* products contain basic operating system software, utilities, and window system software. Extra-cost software options, such as Network File System (NFS) and 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*(1M):

```
% 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.

Patch Releases

A *patch* is a collection of one or more files that correct flaws in the performance, reliability, or security of a specific software product; a *patch release* is a distribution containing one or more patches. Each patch remedies a specific set of reported errors in the version of the product to which the patch applies.

Every patch is identified by a unique number, such as *patchSG1234567*, for example. The number is used in manufacturing to identify the collection of errors that the patch corrects. Typically, the reported errors that result in a patch release occur only under certain operating conditions. For this reason, installing a patch is only necessary if a system or site is experiencing a problem that the patch addresses.

The structure of a patch is similar to that of a software product (see Figure 1-1); however, the subsystems in a patch may span several products. For example, a patch for the FDDI product might contain file replacements for *eoe1*, *eoe2*, and *fddivis* subsystems. Patches are different from standard products in another important way: installing a patch replaces individual subsystem files, not the entire subsystem.

Patches are not dependent on other patches; they rely only on the software components in the original version of the affected product. A patch might be incompatible with other patches, however. Compatibility requirements and installation instructions are described in installable release notes, which you can read from the distribution CD using the *CDgrelnotes* or *CDrelnotes* command. Instructions for reading release notes are included in the CD jewel case insert.

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). The directory on a distribution CD that contains the software is always 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.

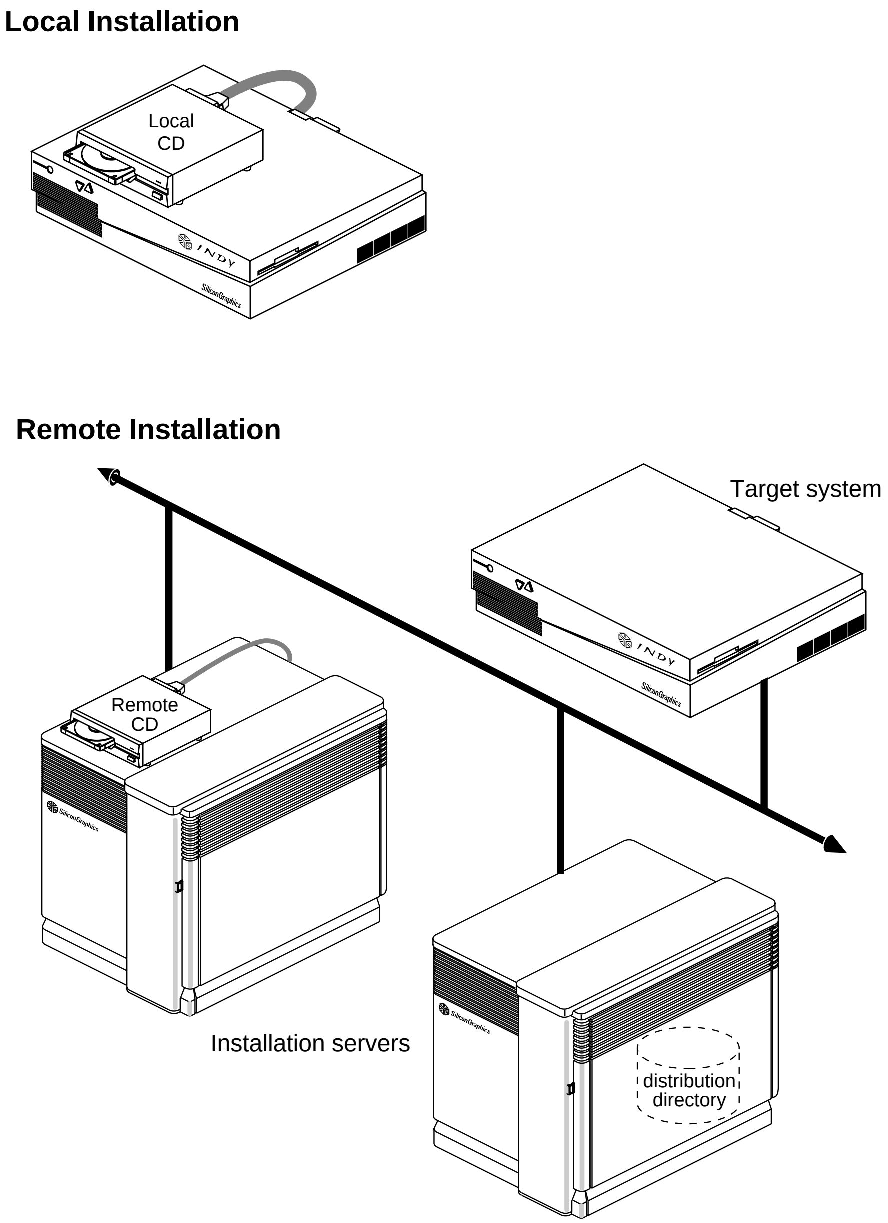


Figure 1-2 Software Distribution Sources

Software Installation Methods

As of IRIX 5.3, most software installations can be performed without interrupting system operations. 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. It is important to understand that a live installation requires more disk space than miniroot installations; during live installations, Inst makes copies of files that are currently in use by the operating system or by applications.

In some cases, software can only 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.

A miniroot installation is required under these conditions:

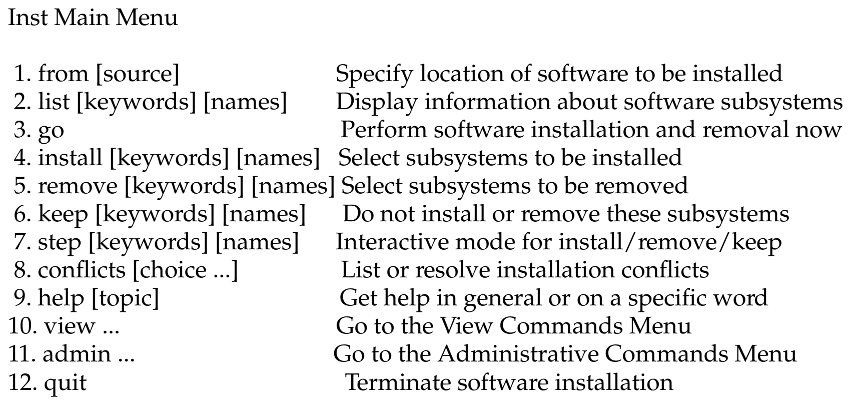
- When the target is being upgraded from a pre-5.3 version of IRIX to IRIX 5.3 or later
- When the release notes state that a miniroot installation is required
- When the target has software problems that prevent successful live installations
- When the target's system disk is new or empty
- When recovering from severe system crashes (system recovery)

Introduction to Inst

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

Note: It might be useful to look ahead at Figure 5-1 for an illustration of the installation process while reading this Introduction.

Figure 1-3 illustrates the Main menu.



```
Inst Main Menu

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

Figure 1-3 The Main Menu for Inst

Automatic Initialization

During initialization, Inst reads the information in product descriptions and the installation database, makes preliminary selections of subsystems to install (see "Required and Default Subsystems" on page 6), and calculates estimated space requirements. Initialization is triggered when you issue the first command of the session that requires Inst to collect information about the distribution or the target software.

Using Main Menu Commands

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 where it is running. Inst selects a default distribution source, which is either the distribution source of the previous installation (during live installations) or local CD-ROM (during miniroot installations). You can specify a different distribution source using the *from* command.

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 images or entire products. 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 the 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 and select software at the same time.

You can change selections as many times as necessary, so 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 when you select two incompatible subsystems for installation, or when you select a subsystem for installation and do not select its prerequisites. 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 clean-up operations on target filesystems and on the newly installed software.

Chapter 2

Preparing for Installation

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

The chapter contains these sections:

- "Selecting a Distribution Source"
- "Setting Up an Installation Server"
- "Backing Up the Target Systems"
- "Planning the Order of Installation"
- "Collecting the Information That You Need"
- "Preparing the Remote CD-ROM Drive"
- "Determining the Installation Method"

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 availability of the server's CD-ROM drive is subject to interruption.

When you are installing software on one or two target systems and the targets contain local CD_ROM drives, using a locally mounted distribution CD is often the most efficient distribution source, particularly if your network is slow and you plan to install a considerable amount 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.

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.

Note: Do not create an installation server on "multi-homed" system (routers with more than one IP address). Packets sent to multi-homed systems are not necessarily returned to the sender by the same route. This can cause problems during miniroot installations.

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. If you plan to use a remote CD-ROM drive as the distribution source, complete all procedures in this section except "Creating a Distribution Directory" 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. On factory-shipped Silicon Graphics systems, however, *bootp*(1M) forwarding is disabled in the file.

Note: The *inetd.conf* file is stored in */usr/etc* on systems running versions of IRIX that are earlier than 5.2.

The procedure below describes how to identify routers and change the */etc/inetd.conf* file to enable *bootp* forwarding on them. This procedure might not be appropriate for routers that are not Silicon Graphics systems.

1. Determine the names of routers.

   Issue this command from the target system:

   `ping -R -c 1` *installation_server*

   Look for an entry that begins with *RR:* in *ping* output. It shows the route of a packet 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.

   ```
   % 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 and 3 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 give the *killall*(1M) command to put the change into effect.

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. However, on factory-shipped Silicon Graphics systems, *tftpd*(1M) access is disabled in the */etc/inetd.conf* file.

This procedure 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 in a way that suits the needs of your site. The default *tftp* entry in */etc/inetd.conf* looks similar to this (your entry might contain additional arguments to the **-s** option):

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

   Take any 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 give the *killall*(1M) 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 83 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/guest/.rhosts* on the installation server contains these lines to permit installations on targets called *joesbox* and *lab1*:

  ```
  joesbox.engr.xxx.com   instuser
  lab1.engr.xxx.com      instuser
  ```

Either of the following commands can be used to install software on *joesbox* or *lab1* (see “Step 2: Specifying the Source” on page 51 and “Specifying the Source on the Command Line” on page 83 for details on these commands):

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

Creating a Distribution Directory

Follow the procedure below to create a distribution directory on the rinstallation server. The CD-ROM drive from which you copy the distribution software may be either a local or remote drive. You can copy several CDs to the distribution directory if you wish; however, do not include more than one release of a given product in the directory—different distribution directories must be created for different releases of a product.

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 that appear in parentheses beside the *scsi* and *cdrom* entries:

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

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/sccntlrdunitl0
   ```

   Note: This command has a lowercase *l* and a zero at the end of it.

4. Create the directory for the distribution.

   ```
   # mkdir distdir
   ```

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

   Issue this command on the system with the CD-ROM drive to stop the *mediad*(1M) daemon:

   ```
   # mediad -k
   ```

6. Create a mount-point directory for the CD-ROM if none exists.

   ```
   # mkdir /CDROM
   ```

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

   ```
   # mount -o ro /dev/dsk/dkscntlrdunits7 /CDROM
   ```

8. Copy the distribution to the new directory.

   On the system containing the distribution directory, issue the *cp* command to copy the installable software from the CD to the distribution directory. The following sample commands illustrate how to copy all distribution software to the directory, creating the necessary subdirectories simultaneously. Use the first command if the CD-ROM is local and the second command if the CD-ROM is remote:

   ```
   # cp -r /CDROM/dist/. distdir
   # rcp -r guest@server:/CDROM/dist/. distdir
   ```

   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/dist/product* distdir/product
   ```

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

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

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

10. Eject the distribution CD.

    Issue 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/sccntlrdunitl0
    ```

11. Repeat steps 7 through 9 for additional CDs.

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

    If you stopped *mediad* in step 5, restart it by issuing 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 requirement for installation, it is strongly recommended. The *Personal System Administration Guide* describes general backup procedures and using System Manager to perform backups. Other programs you can use to make backups are *backup*(1), *bru*(1), *cpio*(1), and *tar*(1). See the reference pages for these programs for more information.

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 that is 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 pre-installation 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:

Note: If you follow these instructions, you will overwrite the installed configuration file. Refer to “Managing Configuration Files” on page 69 for more information on how configuration files are handled during installations.

```
# tar xv files
```

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

```
# tar xv
```

Planning the Order of Installation

Inst automatically manages the installation order of the products on a single CD or in one distribution directory. However, if you expect to install software from more than one CD or distribution directory, you must plan the installation order, since some products require that other products be installed first.

If you are installing from multiple CDs, use the sequence numbers on their labels to put the CDs in order. Install the CDs starting with the lowest sequence number first. Use these guidelines to plan the order of your installation:

- For miniroot installations, be sure that the first CD or distribution directory that you install contains installation tools. CDs containing installation tools are clearly marked.
- If any CD has two sequence numbers, that CD is used twice during the installation. If you find no intervening sequence numbers, you need to insert the CD only once during the installation.
- If you have two or more CDs with the same sequence number, the order of those CDs relative to each other does not matter. For example, assume that you have four CDs with sequence numbers 400, 500, 600, and 600. The CD labeled 400 is installed first, followed by the CD labeled 500. The order of the two CDs labeled 600 is irrelevant, as long as they are installed last.

If you are installing from several distribution directories, check to see what products are in each directory. Plan to install the products in this order:

- Installation tools (for miniroot installations)
- Operating system software
- Communications software
- Compilers
- Optional software

Collecting the Information That You Need

When you install software from a remote distribution source, you are required to supply information about the installation server and target. If your distribution source is a local CD-ROM drive, you may have to supply the controller and unit number of the drive. Jot down the information that you get as you complete these instructions and have it available when you start the installation.

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 are required for miniroot installations when the distribution source is on a remote installation server. Issue the *hostname* command from the target system to get its name:

```
% hostname
```

Issue 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 target_name
```

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 already on the installation server, issue the *hostname* command to get its name:

```
% hostname
```

If you are on the target system, issue this *ping* command to get the Internet Protocol (IP) address of the installation server:

```
% /usr/etc/ping -c 1  installation_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
```

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 A, "Installation Troubleshooting" 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 19).

Preparing the Remote CD-ROM Drive

You can use a utility called *cdinstmgr* to manage the mounting and unmounting of CDs on a remote CD-ROM drive during installations. This utility prevents other users from ejecting distribution CDs during an installation and informs you of error conditions. You must start *cdinstmgr* before you begin the installation.

Use this procedure to start *cdinstmgr*:

1. Become root and issue the *cdinstmgr* command on the installation server.

   ```
   % su
   # cdinstmgr
   ```

2. Respond to *cdinstmgr* prompts.

   Follow *cdinstmgr* directions as it prompts for the name of the target and guides you through the mechanics of loading the first CD. Be sure to insert CDs with the label up. If *mediad*(1M) is not already running on the installation server, *cdinstmgr* prompts you for the superuser password to the server.

3. Note the mount directory for the CD-ROM drive.

   When you see the message below, note the value of *CDdir*,the mount point directory; you will need it later.

   ```
   CD on CDdir for host hostame: type the word "done" when you
   are finished with this CD, "quit" if you are completely
   done:
   ```

   You will respond to this prompt after you have finished using this CD.

4. Verify that the target can access the distribution source.

   Issue this command from the target to verify that it has access to the distribution CD:

   ```
   % rsh server -l user 'ls CDdir/dist'
   ```

   If the output to this command is a list of products contained on the distribution CD, the target has access to the distribution CD. If the output is an error message, you should diagnose and correct the error before beginning the installation.

Determining the Installation Method

Software products containing subsystems that require a miniroot installation cannot be installed by a live installation (see “Software Installation Methods” on page 10). To install these subsystems, you must shut down the target system and install them from the miniroot. 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.

If you determine that a miniroot installation is required, complete the procedures in Chapter 3, “Starting a Miniroot Installation” and Chapter 5, “Using Inst” to perform the installation (the procedures in Chapter 4 are not used during a miniroot installation).

If you determine that a live installation is suitable, complete the procedures in Chapter 4, “Starting a Live Installation” and Chapter 5, “Using Inst” to perform the installation (the procedures in Chapter 3 are not used during a live installation).

Chapter 3

Starting a Miniroot Installation

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

This chapter explains how to load the miniroot to start a miniroot installation. If you determine that a miniroot installation is necessary (see "Determining the Installation Method" on page 27), you must complete the procedures in this chapter before going on to Chapter 5, "Using Inst."

This chapter contains these sections:

- "Understanding Miniroot Installations"
- "Before You Begin a Miniroot Installation"
- "Preliminary PROM Monitor Procedures"
- "Loading From a Local CD"
- "Loading From a Local CD"
- "Finishing the Miniroot Load"

Note: General instructions for using the PROM monitor are provided in this chapter. For PROM monitor information that is specific to a particular target model, refer to these publications:

- The *Advanced Site and Server Administration Guide*. See the chapter dedicated to the PROM monitor.
- The owner's guides for Onyx and Challenge systems (see the power-on instructions for PROM monitor information) and for personal workstations, such as Indigo and Indy models (see the software installation instructions for PROM monitor information).

Understanding Miniroot Installations

During a miniroot installation, the miniroot is transferred from the distribution source to the swap partition of the target workstation in a process known as *loading the miniroot*. The miniroot is loaded by a program called the *sash*, or standalone shell, which is a component of the installation tools that accompany some products (see Figure 1-1).

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 and another partition is used for swap, the miniroot installation will still occur in partition 1 and overwrite any data in that partition. In this case be sure to back up any data in partition 1 prior to the miniroot installation.

The miniroot contains an IRIX kernel, Inst, and other tools and files. 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.

To access the installation tools, you use the *PROM monitor*, the user interface to the programmable read-only memory (PROM) of the target server or personal workstation. The commands that you enter using the PROM monitor load the miniroot and automatically invoke Inst. The PROM monitor features menus and prompts and a command line interpreter, the *command monitor*, which supports a small set of commands for performing simple functions.

The user interface to the PROM monitor varies slightly on different models of Silicon Graphics computers. For example, on graphics systems, some PROM monitor menus contain icons that you click to make selections. If you are using an older model system or an ASCII console on a graphics system, all prompts and menus appear as text. But for installation purposes, the tasks that you perform from the PROM are the same for all computer models, regardless of differences in the user interface.

Before You Begin a Miniroot Installation

If you have not already done so, complete these preparations before beginning a miniroot installation session:

1. Check product release notes for special installation requirements.
2. Complete network preparations if the software distribution is remote.
   - 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 18 for instructions).
   - Verify that the guest account or an alternate account is available to Inst on the remote distribution server ("Configuring an Installation Account" on page 19 for information).
3. Back up the target system.

   Backing up essential files on the target system is not normally required but it is strongly recommended (see "Backing Up the Target Systems" on page 22 for instructions).
4. Jot down the information that you will be required to supply.
   - Be prepared to compare the IP address of the target with the address setting in the target's PROM (see "Getting the Target's Name and Address" on page 25 for instructions on determining IP addresses).
   - 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 24 for instructions).
   - If your distribution source is on a remote installation server, be prepared to specify the server name and IP address, as well as the pathname of the distribution directory or CD-ROM directory on the server (see "Getting the Server's Name and Address" on page 25 for instructions).

5. Plan the software installation order.

   If you are installing from more than one CD or distribution directory, be sure to plan the order of installation (see “Planning the Order of Installation” on page 23 for instructions).

Preliminary PROM Monitor Procedures

The procedure in this section is done from the PROM monitor on the target to prepare the target for miniroot loading. The procedure has three objectives:

- To verify the target’s network address

  For remote miniroot loads, the address that is set in the target’s PROM must be identical to its IP address (see “Getting the Target’s Name and Address” on page 25). Differences in these address settings cause the miniroot load to fail. In most cases, the two addresses are the same. However, as a safety precaution, it is best to compare the target’s IP address with its PROM network address and change the PROM address if it is different.

- To determine the processor type for the target

  During miniroot installations on some targets, you may need to specify the type of the central processor unit (CPU) in the target system. You select the appropriate procedure for loading the miniroot onto the target based on its CPU type. You can determine the CPU type (CPU model numbers begin with the letters *IP*) by issuing the *hinv* or *version* command from the command monitor or the *hinv*(1M) command from an IRIX shell.

- To change the PROM’s default setting of the distribution source, if needed

  On server and personal workstation models containing an IP4, IP5, IP6, IP12, and IP17 processor, the default distribution source for an installation is the local tape device. If you plan to do a remote installation, this default must be reset (if you have a tape drive installed) so that you can specify a remote distribution source.

Follow these instructions to complete the preliminary PROM procedures:

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:

   ```
   % su –
   Password:
   # shutdown
   ```

   After the shutdown, you see this prompt:

   ```
   Okay to power off the system now.
        Press any key to restart.
   ```

   Note: If you do not see the previous message after the target shuts down, it means that the target contains an IP4 processor, which does not support the menu features described in steps 2 through 4. Make a note of the IP number and go directly to step 5 of this procedure.

2. Begin the restart sequence.

   Press any key (such as <*Enter*>) to begin the system restart, but do not allow the restart to complete; go immediately to step 3.

   Note: If the system restarts before you are able to complete step 3, repeat steps 1 and 2 of this procedure.

3. Go to the System Maintenance menu.

   On systems without ARCS PROMs, you see the text message shown below. Type <*Esc*> to go to the System Maintenance menu when you see this message:

   ```
   Starting up the system...
   To perform system maintenance instead, press <Esc>.
   ```

   On systems with ARCS PROMs, you see the *Stop for Maintenance* button. Click the *Stop* button or press **<Esc>** to go to the System Maintenance menu.

4. Choose *Enter Command Monitor* from the System Maintenance menu.

   On newer systems (those with ARCS PROMs), the System Maintenance menu is similar to the one shown in Figure 3-3. Click the *Enter Command Monitor* icon on this menu.

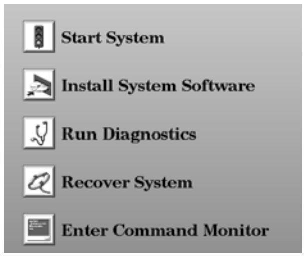


Figure 3-1 System Maintenance Menu

On older systems (which do not contain ARCS PROMs), the System Maintenance menu is similar to the one shown Figure 3-2. Type the number of the *Enter Command Monitor* selection. In Figure 3-2, the number of the entry is 5.

```
System Maintenance Menu

1)  Start System
2)  Install System Software
3)  Run Diagnostics
4)  Recover System
5)  Enter Command Monitor

Option? 5
```

Figure 3-2 System Maintenance Menu on Older Systems

After your entry, the command monitor prompt is displayed. It looks similar to the one shown here:

```
Command Monitor. Type “exit” to return to the menu.
>>
```

5. Use the *printenv* command to display PROM variable settings.

```
>> printenv
```

The output of *printenv* shows the PROM setting for the network address as the value of *netaddr*, as shown in this example:

```
netaddr=123.45.6.78
```

If the value of *netaddr* is different from the target address that you obtained from the *ping* command (see "Getting the Target's Name and Address" on page 25), reset the PROM address to the address that you obtained from *ping*. Use this command to set the address:

```
>> setenv netaddr ping_address
```

6. Use the *hinv* command (or the *version* command if you do not have *hinv*) to display the CPU number.

   ```
   >> hinv
   ```

   The *hinv* command provides the CPU type of the target. Scan the output of *hinv* for a number that starts with the letters *IP*, which is the CPU type of the target. The line looks like one of these examples:

   ```
   System: IP22
   ```

   ```
   CPU board: IP12 33 MHz, with FPU
   ```

   Make a note of the CPU type.

   Note: If the CPU type shown by *hinv* is IP15, the CPU is actually an IP7.

7. Reset the default distribution source, if needed.

   If you plan to load the miniroot from a remote distribution source and you determined that the target's CPU is an IP4, IP5, IP6, IP12, or IP17, (and you have a tape drive) reset the default distribution source. Use this command to reset the default source (which is local tape):

   ```
   >> setenv notape 1
   ```

The procedure that you use to load the miniroot depends on the location of your distribution source (local or remote) and on the CPU type of the target contains. Use one of these procedures to continue:

- If you plan to use a local CD as the distribution source, use the instructions in "Loading From a Local CD" for all CPU types.
- If you plan to use a remote distribution source, go on to "Loading From a Local CD" on page 36. Select the procedure from this section that is appropriate for the type of CPU that the target system contains.

Loading From a Local CD

This procedure explains how to load the miniroot from a distribution CD that is mounted locally on the target. Because of PROM monitor variations, the procedure for loading the miniroot on IP4, IP5, IP6, IP7, IP9, or IP17 targets is somewhat different from the procedure on other systems.

Note: If you determined from “Preliminary PROM Monitor Procedures” on page 32 that the target contains an IP4, IP5, IP6, IP7, IP9, or IP17 processor, do not complete steps 1 and 2 of this procedure. Go directly to step 3 to begin the miniroot load.

1. Exit the command monitor.

   ```
   >> exit
   ```

   The System Maintenance menu is displayed after your entry.

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

   On most systems, the System Maintenance menu is similar to the one shown in Figure 3-3. Click the *Install System Software* icon on this menu.

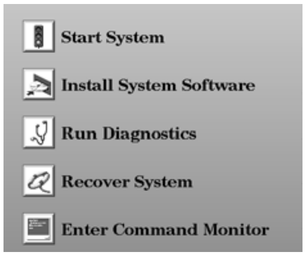
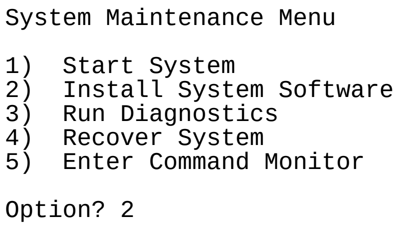


Figure 3-3 System Maintenance Menu

   On older systems (which do not contain ARCS PROMs), the System Maintenance menu is similar to the one shown in Figure 3-4. Type the number of the *Install System Software* selection (in Figure 3-4, the number is 2).



```
System Maintenance Menu

1)  Start System
2)  Install System Software
3)  Run Diagnostics
4)  Recover System
5)  Enter Command Monitor

Option? 2
```

Figure 3-4 System Maintenance Menu on Older Systems

3. Select the local CD-ROM icon (not required on older systems).

   On systems with ARCS PROMs, a menu of distribution sources is displayed after you choose *Install System Software*. Click the icon for the local CD-ROM to continue with the installation.

   If you are using an ASCII console, the menu of distribution sources appears as text. To continue, enter the number of the Local CD-ROM selection.

   On older systems, this menu does not appear; omit this step.

4. Insert the first CD.

   Be sure to insert the CD containing the installation tools into the CD caddy with the label side up. Then insert the caddy into the CD-ROM drive.

5. Load the miniroot and invoke Inst.

   If the target contains an IP4, IP5, IP6, IP7, IP9, or IP17 (or an IP12 other than the Indigo) processor, enter a command like the example shown below to load the miniroot and invoke Inst. Substitute the CD-ROM controller number (*cntlr*), unit number (*unit*), and CPU number (*cpu*) in your entry:

   ```
   >> dksc(cntlr,unit,8)sash.cpu -m
   ```

   Note: Only one blank space is entered in the previous command; it follows *cpu*.

   On targets containing any other CPU type, click the *Continue* button or press <*Ent*>. These messages appear after your entry:

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

As the miniroot load is completed, additional messages are displayed. Go to "Finishing the Miniroot Load" on page 42 for information on interpreting these messages.

Loading From a Remote Distribution Source

The model number of the CPU in the target determines the procedure that you follow to load the miniroot from any remote distribution source (CD or distribution directory). This section contains three procedures for loading the miniroot based on CPU number:

- "Remote Loading on an IP4, IP5, IP7, or IP9"
- "Remote Loading on an IP6, IP12, or IP17"
- "Remote Loading on an IP19, IP20, or IP22"

Select the procedure that is appropriate to the installation target.

Remote Loading on an IP4, IP5, IP7, or IP9

If you determined that the target's CPU number is an IP4, IP5, IP7, or IP9 (see "Preliminary PROM Monitor Procedures" on page 32), you can load the miniroot from the command monitor. The commands that you use to perform the load depend on the location of your distribution source.

If your distribution source is a remote CD, enter the commands below to load the miniroot and invoke Inst. Substitute specific information in your entries*:* *installation_server* is the hostname of the server containing the CD, *CDdir* is the mount directory of the CD-ROM drive, and *cpu* is the target's CPU number.

```
>> setenv tapedevice bootp()installation_server:CDdir/dist/sa
>> boot -f $tapedevice(sash.cpu) --m
```

If your distribution source is a remote directory, enter the commands shown below to load the miniroot and invoke Inst. Substitute specific information in your entries*:* *installation_server* is the hostname of the server containing the distribution directory, *distdir* is the pathname of the distribution directory, *sa* is the command that starts the load, and *cpu* is the target's CPU number.

```
>> setenv tapedevice bootp()installation_server:distdir/sa
>> boot -f $tapedevice(sash.cpu) --m
```

Messages are displayed as the miniroot load is completed. Go to "Finishing the Miniroot Load" on page 42 for information on interpreting these messages.

Remote Loading on an IP6, IP12, or IP17

If you determined that the target's CPU is an IP6, IP12, or IP17 (see "Preliminary PROM Monitor Procedures" on page 32), use this procedure to load the miniroot and invoke Inst on the target. When you complete this procedure, go on to the instructions in Chapter 5, "Using Inst," to install software.

1. Exit the command monitor.

   ```
   >> exit
   ```

   After you exit the command monitor, the System Maintenance Menu is displayed, as shown in Figure 3-5.

```
System Maintenance Menu

1)  Start System
2)  Install System Software
3)  Run Diagnostics
4)  Recover System
5)  Enter Command Monitor

Option?
```

Figure 3-5 System Maintenance Menu on Older Systems

2. Select *Install System Software* from the System Maintenance menu.

   Type **2** to install system software. After your entry, you see messages like the ones below:

   ```
   Installing System Software...
   Press <Esc> to return to the menu.
   Are you using a remote tape? (y/n) n
   ```

3. Answer no (**n**) to the remote tape prompt.

   After your entry, you see a prompt like the one below:

   ```
   Enter the hostname of the server that has the
   installation software (e.g. hostname:directory):
   ```

4. Load the miniroot and invoke Inst.

   If your distribution source is a remote CD, enter the command shown below to load the miniroot and invoke Inst. Substitute specific information in your entry: *installation_server* is the hostname of the server containing the CD, and *CDdir* is the mount directory of the CD-ROM drive:

   installation_server`:`*CDdir*`/dist`

   If your distribution source is a remote directory, enter the command shown below to load the miniroot and invoke Inst. Substitute specific information in your entry: *installation_server* is the hostname of the server containing the distribution directory, *distdir* is the pathname of the distribution directory, and *cpu* is the CPU number of the target:

   installation_server`:`*distdir*

Messages are displayed as the miniroot load is completed. Go to “Finishing the Miniroot Load” on page 42 for information on interpreting these messages.

Remote Loading on an IP19, IP20, or IP22

If you determined that the target’s CPU is an IP19, IP20, or IP22, use this procedure to load the miniroot and invoke Inst on the target. When you complete this procedure, go on to the instructions in Chapter 5, “Using Inst,” to install software.

1. Exit the command monitor.

   ```
   >> exit
   ```

   After your entry, the Maintenance menu is displayed, as shown in Figure 3-3.

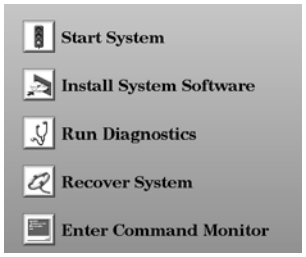


Figure 3-6 System Maintenance Menu

2. Select *Install System Software* from the System Maintenance menu.

   After your selection, a menu of distribution sources is displayed. On an ASCII console, the menu of distribution sources appears as text.

3. Select *Remote Directory* as the distribution source.

   Click the *Remote Directory* icon to select the distribution source. If you are using an ASCII console, enter *2*.

4. 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
   ```

5. 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 remote CD, 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 remote directory, your entry resembles the example shown below. Substitute the pathname of the distribution directory for *distdir*:

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

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

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

6. 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:CDdir/dist
   Copying installation tools to disk
   ```

Additional messages are displayed as the miniroot load is completed. Go to "Finishing the Miniroot Load," which follows, for information on interpreting these messages.

Finishing the Miniroot Load

Use this procedure to finish the miniroot load on all system models, from any distribution source.

1. Wait for the miniroot to be copied.

   Copying the miniroot to the swap partition of the target takes several minutes. During that time, rows of dots appear, indicating the status of the copy (a message bar appears instead on systems with ARCS PROMs). 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 appears for each type of device supported by the kernel, but not present on the target. After these 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.
```

2. Reset the system date, if needed.

   Notice that the startup messages included 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 Timeout Value” on page 124 if you need more information):

   ```
   Inst> admin date mmddhhmmyy
   ```

   You can display the date again with this command:

   ```
   Inst> admin date
   ```

3. 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 (shown in Figure 5-2):

   ```
   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 on to the procedures in Chapter 5, “Using Inst” to install software.

Chapter 4

Starting a Live Installation

This chapter describes how to invoke Inst for a live installation. It also explains what to do to prepare for live installations. Use the procedures in this chapter to perform any installation that does not require the miniroot.

This chapter contains these sections:

- "Before You Begin a Live Installation"
- "Invoking Inst for Live Installations"

Note: Do not perform the procedures in this chapter if you completed the procedures in Chapter 3.

Before You Begin a Live Installation

If you have not already done so, complete these preparations before beginning a software installation session:

1. Check product release notes for special installation requirements.
2. Complete network preparations if the software distribution is remote.

   The guest account or an alternate account must be available to Inst on the remote distribution server (see "Configuring an Installation Account" on page 19 for instructions).
3. Back up the target system.

   Backing up essential files on the target system is not required but it is strongly recommended (see "Backing Up the Target Systems" on page 22 for instructions).

4. Make a note of the exact location of the software distribution.

   If your distribution source is on a remote host, be prepared to specify the remote hostname and pathname of the distribution directory or CD-ROM drive during the installation (see “Getting the Server’s Name and Address” on page 25 for instructions).

5. Plan the software installation order.

   Inst manages the order automatically when you install from a single CD or distribution directory. But if you are installing from more than one CD or distribution directory, be sure you have planned the order of installation (see “Planning the Order of Installation” on page 23 for instructions on installation order).

Note: Live installations normaly require extra temporary disk space. During a live installation, Inst must create copies of some files that are held open by the operating system or by applications.

Invoking Inst for Live Installations

To install software, you must invoke Inst as the superuser. If you invoke Inst as a regular user, you are assigned read-only permissions, which prevents you from making any changes to the installed software.

Use this command to invoke Inst from the command line of an IRIX shell:

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

After your entry, the default location of the distribution is posted and the Inst Main menu is displayed (see Figure 5-2).

Note: When doing 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.

Chapter 5

Using Inst

This chapter describes how to install software using the commands available from the Inst Main menu. It explains how to use Inst commands in each step of a typical installation procedure and what to do after the installation is completed to put the target back into service. After you have performed the procedures in Chapter 3, “Starting a Miniroot Installation” or Chapter 4, “Starting a Live Installation,” you can use the instructions in this chapter to complete an installation.

This chapter contains these sections:

- “Getting Help During Installations”
- “The Software Installation Procedure”
- “Stopping the Remote CD Manager”
- “Managing Configuration Files”
- “If Inst Is Interrupted”

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 to do the installation (for live installations only). See the software installation instructions in the Personal System Administration Guide for help with using Software Manager.

Getting Help During Installations

During an installation session, you can use the *help* command to get information about almost every command and command argument that appears on Inst menus. 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 9-1 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 very important to using Inst successfully. To get this information, issue this command during an installation session:

```
Inst> help overview
```

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 is somewhat arbitrary; it 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 issue commands from other menus during installation sessions.

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

1. Invoke Inst.
2. Specify the distribution source using the *from* command.
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.

Note: Inst never installs or removes any software until you enter the *go* command.

Figure 5-1 illustrates the steps in the installation process. Notice that Inst performs an automatic initialization sequence during the process. Initialization is triggered by the first command of a session that requires Inst to collect information about the distribution or target software. In Figure 5-1, the trigger is the *from* command.

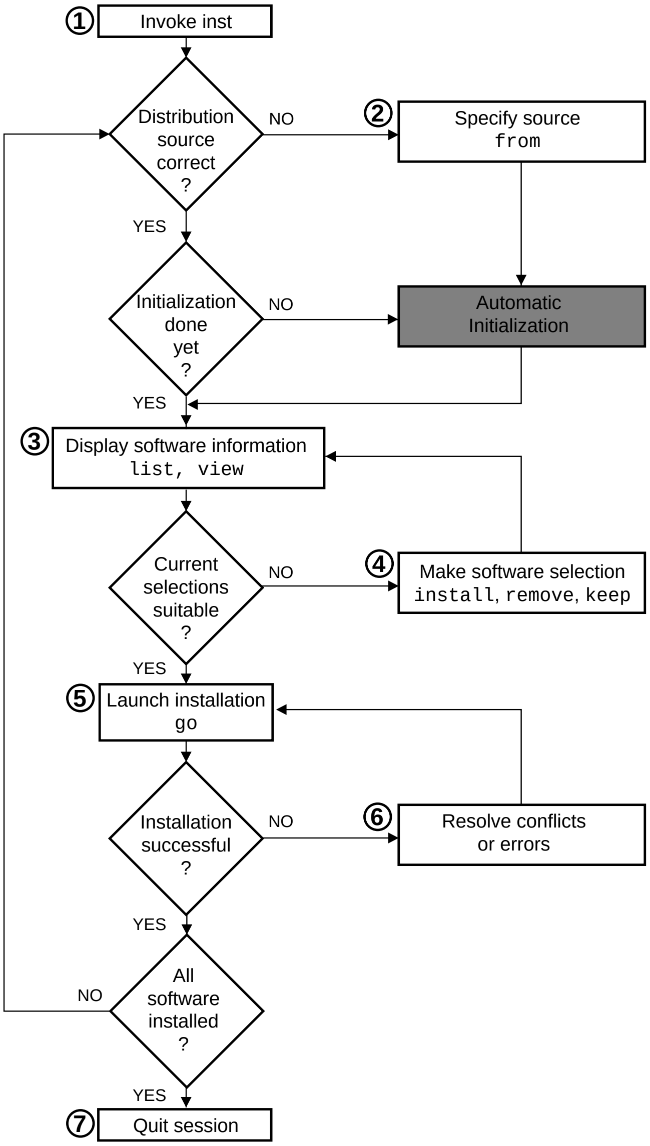
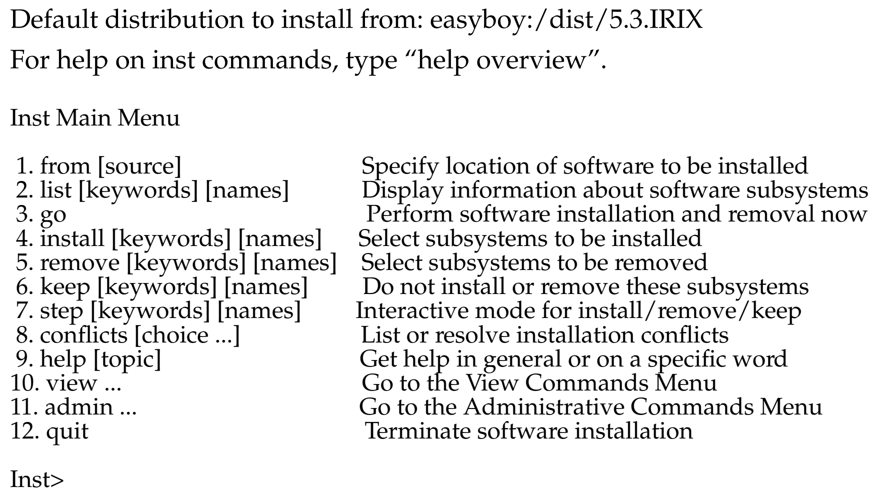

1
Invoke inst
Distribution source correct ?
NO
2
Specify source from
YES
Initialization done yet ?
NO
Automatic Initialization
YES
3
Display software information list, view
Current selections suitable ?
NO
4
Make software selection install, remove, keep
YES
5
Launch installation go
Installation successful ?
NO
6
Resolve conflicts or errors
YES
All software installed ?
NO
YES
7
Quit session

Figure 5-1 Summary of the Installation Procedure

Step 1: Invoking Inst

For miniroot installations, Inst is automatically invoked when the miniroot is loaded (see “Finishing the Miniroot Load” on page 42). For a live installation, Inst is invoked from the command line of an IRIX shell (see “Invoking Inst for Live Installations” on page 46). After Inst is invoked, tit displays the default distribution source and the Main menu, as shown in Figure 5-2.



```
Default distribution to install from: easyboy:/dist/5.3.IRIX
For help on inst commands, type “help overview”.

Inst Main Menu

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

Inst>
```

Figure 5-2 The Inst Main Menu

Step 2: Specifying the Source

Perform this step only if you need to change the distribution source from which software will be installed. 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 invoked. You can either accept the default source or change it using the *from* command.

Using the from Command

Use the *from* command to change from one distribution source to another. You can issue the *from* command to change distributions as many times as you need to during an installation session. Your entry can take one of the forms of *from* listed below (also see "Alternative Ways to Specify a Distribution" on page 82):

`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 is not posted, enter an alternate form of the *from* command to specify the distribution.

`from` *pathname*

Add a pathname argument to *from* to specify a distribution that is local to the target system (usually a CD-ROM drive). Do not use the tilde (~) in your specification.

These examples illustrate how to specify distributions that are on a local CD-ROM drive and on a local disk directory:

```
from /CDROM/dist
from /d1/IRIX_5.3/dist
```

This form of *from* command also applies to directories and CD-ROM drives that are NFS mounted on the target system.

`from` *server*:*pathname*

Add the server name and pathname arguments, separated by a colon, to specify a distribution that is located on a remote system. These examples illustrate how to specify a distribution that is located on a remote directory and remote CD-ROM drive:

```
from easyboy:/d1/IRIX_5.3/dist
from easyboy:/CDROM/dist
```

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.

Automatic Initialization Functions

Before Inst takes any installation actions, it reads information about the distribution and target. 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 and calculates their sizes, and checks disk availability on the target. As this initialization occurs, you see messages similar to these:

```
Reading installation history .. 100% Done.
Reading distribution .. 100% Done.
Checking dependencies .. 100% Done.
Calculating sizes .. 100% Done.
```

When the automatic functions are finished, the `Inst>` prompt appears, and you can continue with the installation session.

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

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 plan to install. In most cases, however, this step is necessary to evaluate distribution and target software and to 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 target software using the *view* command (see “Using the View Commands Menu” on page 92 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 issue 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 84 for details).

Example 5-1 illustrates a *list* entry and the legend that preceeds *list* displays.

```
Inst> list

Legend:
Current View:
  Location:  distribution
  Status:    N=New,U=Upgrade,P=Patch upgrde,S=Same,D=downgrade,' '=Not Installed
  Patches:   A=patch installable,X=patch uninstallable (missing base product)
  Selection: i=install, r=remove, k=keep
  Level:     subsystem
  Name:      short

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

Example 5-1 The list Legend

The *list* legend identifies the location of the software being displayed (*distribution* in Example 5-1), a description of the mnemonic tags that provide information about the software components in the display (listed in the *Status*, *Patches*, and *Selection* lines), the level of the product hierarchy that is displayed (*subsystem*), and the sorting order of the displayed products (sorted alphabetically by product *short*names in the example).

Example 5-2 illustrates sample *list* output of a distribution inventory.

```
  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
i U compiler_eoe.sw.lib [bdr]     3+  Base Execution Libraries
i U compiler_eoe.sw.unix         40+ IRIX Execution Environment

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

  N onc3_eoe.sw.cachefs           0  ONC3 5.3 Cache File System
```

Example 5-2 Sample list Display of Distribution Software

Frequently, *list* displays comprise several screens of information. To display more information, type <*Enter*> at the *more* prompt for a new line and <*Space*> for a new screen. You can also quit the display before reaching the end by typing *q* at the *more* prompt.

Displaying Target Software

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

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

Example 5-3 shows sample *list* output of target software (the legend is omitted from this display).

```
  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
u I compiler_eoe.sw.lib [bdr]    3- Base Execution Libraries
u I compiler_eoe.sw.unix        40- IRIX Execution Environment

  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
```

Example 5-3 Sample list of Target 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 5-1). Columns appear empty when a tag is unassigned or when the tag is masked by the current view setting (see “Using the View Commands Menu” on page 92 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.

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 5-1).

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 space is freed. 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 5-2 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. Since 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 5-3 indicates that most products from the distribution are currently installed (I) on the target, but the *showcase* subsystems have been removed (R). One distribution product from Example 5-2, *onc3_eoe*, does not appear in the target list in Example 5-3. Also notice that items marked for installation (i) in the distribution list are marked for upgrading (u) in the target list.

Interpreting Patch Information

A *list* display also includes patch components if patches are present on the distribution or target. Example 5-4 illustrates patch entries in a *list* display.

```
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
```

Example 5-4 Sample list of Patches

The mnemonic tags on patches have these meanings:

DA (Downgrade Available) This patch applies to an installed base subsystem; it is a downgrade to another patch that is already installed.

NA (New Available) 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 upgrades installed software.

SA (Same Available) This patch is already installed.

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 5-5 (summary from a live installation session) and Example 5-6 (summary from a miniroot installation). 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, and displays the amount of space available on the target filesystems.

```
Disk Space summary (Kbytes)         /
Selections net change            1276+
Space available                 564432
```

Example 5-5 Sample Disk Space Summary for Live Installations

```
Disk Space summary (Kbytes)   /     /root     /root/usr
Selections net change         0        2+          149+
Space available           11984      6062       216876
```

Example 5-6 Sample Disk Space Summary for Miniroot Installations

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 capacity of the target and the needs of its users. However, in some cases it is possible to accept the automatic selections that Inst made during initialization. These selections are described in “Accepting Automatic Selections” on page 62.

Note: If the automatic selections described on page 62 are suitable for your circumstances, you can accept these selections and go directly to step 5 of this procedure.

Using 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 issue the *go* command (see “Step 5: Launching the Installation” on page 62), so you can change your selections as many times as necessary before entering *go*.

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 89 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 62). Use the *install* command to select software that is not already selected.
- In some cases, you might not want an automatic (or user) selection installed on the target, or the currently installed version may be more suitable. 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 by software from the distribution (illustrated by Example 5-3). 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 this step (see "Step 3: Displaying Software Information" on page 53).
- It is sometimes useful to alternate the selection process with the launch process (see "Step 5: Launching the Installation" on page 62), selecting and installing a subset of software in each cycle.

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 *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> remove InPerson.books
```

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

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

Retain the *DeltaCC.sw.backend* subsystem (if it is installed) or prevent installation of the distribution version.

You can use multiple arguments in *install*, *remove*, or *keep* commands, as shown in Example 5-7. Your arguments can extend to new lines, as long as the last character on the line is not an empty space.

```
Inst> install sysmon cadmin mmail
Inst> keep InPerson.books InPerson.sw
Inst> remove DeltaCC.sw.backend DeltaCC.man.relnotes
```

Example 5-7 Multiple Arguments to install, remove, and keep

Using Arguments in Selection Commands

You can use the names of products and their components, metacharacters, and keywords as arguments to *install*, *remove*, and *keep* commands to save time (see “Using Command Shortcuts” on page 84 for more information). These sample entries illustrate the use of keywords in selection commands:

```
Inst> install required
```

Install all subsystems that are required for optimum system operation.

```
Inst> remove *.man
```

Remove all images with a *.man* extension from the target.

```
Inst> keep I eoe1*
```

Keep all *eoe1* subsystems that are currently installed on the target (do not install upgrade versions from the distribution).

Checking Your Selections With list

You can issue 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 reflects your selections and their effects on disk space.

Accepting Automatic Selections

Recall from step 3 that Inst automatically made some preliminary selections during initialization (also see "Automatic Initialization Functions" on page 53). If you accept the automatic selections, you can omit step 4 and go on to "Step 5: Launching the Installation" on page 62.

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

```
Inst> list i
```

Example 5-8 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.

```
i  U eoe1.sw.gfx_lib bdr      19+   Graphics Execution Libraries
i  U eoe1.sw.irix_lib bdr     991-  IRIX Execution Libraries
i  U eoe1.sw.unix bdr         6674- IRIX Execution Environment
i  U eoe2.sw.cdrom d          241-  CD-ROM Support
i  U eoe2.sw.crypt d          25-   Security Administration Utilities
i  U eoe2.sw.gltools d        80-   Graphics Library Tools

Disk space summary (Kbytes):            /
Selections net change               64793-
Space available                    538102
```

Example 5-8 List of Subsystems Selected for Installation

Step 5: Launching the Installation

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

Note: Inst will detect a conflict condition if you fail to include any prerequisite software in a selected subset (see "Step 6: Resolving Conflicts and Other Errors" on page 64 for details).

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.

```
Inst> go

Pre-installation check ..  100% Done.
Installing/removing software .. 0%
Installing new versions of selected pv.man subsystems
Installing/removing software .. 44%
Installing new versions of selected pv.sw subsystems
Installing/removing software .. 100% Done.
Removing orphaned directories
Done.

Running exit commands .. 100% Done.
Checking dependencies .. 100% Done.
Installations and removals were successful.
You may continue with installations or quit now.

Inst>
```

Example 5-9 Successful Installation Messages

Notice that Example 5-9 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 67. If you receive an error message instead of a success message, complete "Step 6: Resolving Conflicts and Other Errors" before going on to step 7.

Step 6: Resolving Conflicts and Other Errors

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. Other errors that occur during *go* command processing are resolved with routine corrective actions (see Appendix A, "Installation Troubleshooting," 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 issue the *conflicts* command periodically during the selection process to display any conflicts that might be accumulating as a result of your selections.

Example 5-10 illustrates a message describing two conflicts.

```
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

eoe1.sw.dlpi (DLPI Execution Environment), eoe1.sw.dlpi
(DLPI Execution Environment) and eoe1.sw.dlpi (DLPI
Execution Environment) cannot be removed because other
products depend on them:
  2a. Do not remove eoe1.sw.dlpi (DLPI Execution
Environment),
      eoe1.sw.dlpi (DLPI Execution Environment) and
eoe1.sw.dlpi
      (DLPI Execution Environment)
  2b. Also remove 4DDN.sw.4DDN (4DDN Software)
```

Example 5-10 Sample Conflicts Message

To resolve a conflict, enter the *conflicts* command and your choice of resolutions as command arguments, as shown in Example 5-11.

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

Example 5-11 Sample Entry to Resolve Conflicts

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 5-9), you can continue with the session or go on to “Step 7: Quitting the Installation Session” on page 67.

Correcting Other Installation Errors

If Inst is unable to execute the *go* command 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 pre-installation checking, while software is being installed and removed, or during exit operations.

Example 5-12 illustrates an error that occurs during pre-installation checking.

```
Inst> go

nfs-mounted filesystem /usr/local/bin/ptools is read-only

nfs-mounted directory /usr/local/bin/ptools/app-defaults is
write-protected

nfs-mounted directory /usr/local/bin/ptools is
write-protected

Installations and removals canceled

Inst>
```

Example 5-12 Sample Error in Pre-Installation Processing

Most error messages contain information that suggests the corrective action that is required; however, if you need additional information to correct an error, see Appendix A, "Installation Troubleshooting."

Example 5-13 illustrates an error that occurs during 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
/usr/people/swmgr/products/prod1/version01/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>
```

Example 5-13 Sample Error in Installation Processing

Notice from Example 5-13 that if you encounter errors during installation, Inst invokes the Interrupt menu (explained in "Using the Interrupt Menu" on page 100). This version of the Interrupt menu includes a "retry" choice, which might be used, for example, if the error is caused by a network timeout. If you need additional information to correct the error, see Appendix A, "Installation Troubleshooting."

Example 5-14 illustrates an error that occurs during exit operations.

```
ERROR : Sub-command “cd $rbase/usr/lib/filetype; ./tagscript
$rbase; rm tagscript” returned status 1
100% Done.
Errors occurred executing exit-ops
```

Example 5-14 Sample Error in Exit Operations

See Appendix A, “Installation Troubleshooting” for an explanation of messages that are generated during exit operations.

Step 7: Quitting the Installation Session

You can leave an installation session at any time by issuing the *quit* command at the Inst prompt. If any installation or removal selections are pending, Inst notifies you:

```
Inst> quit
```

```
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 decide to continue the session.

If the installation installed a new icon in the icon database, you see the message:

```
The Icon Database is being updated. Desktop icons may
disappear for a moment.
```

The icons disappear and then reappear after about five seconds.

Handling Unresolved Conflicts

In some cases, there may be unresolved conflicts when you issue the *quit* command to end a session. 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 issue the *quit* command, Inst posts this message:

```
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).
```

Example 5-15 Conflicts Message When Quitting Inst

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 and Other Errors” on page 64 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 5-15. 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 install products that require a system reboot to put the software into effect, you see the message in Example 5-16.

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

Inst>
```

Example 5-16 Quit Prompt With the Reboot Message

After you reboot the system, you see this message:

```
Requickstarting necessary files
```

The Requickstart Sequence

During exit processing, Inst synchronizes the libraries and executable files in the new installation that rely on shared objects. This process is known as a *requickstart*; it ensures a faster startup time for the affected files. If this is the first time that you installed products that require a quickstart, you see the message shown in Example 5-17.

```
1. Continue with the session
2. Quit now
Please enter a choice [1]: 2

Building a one time quickstart data file
Requickstarting necessary files
Automatically reconfiguring the system.
Ready to restart the system. Restart?
(y)es,(n)o,(sh)ell,(h)elp}:
```

Example 5-17 Quit Prompt With a Quickstart File Message

For some installations, the requickstart might require as long as 20 minutes; however, for most installations, much less time is required. During the requickstart, you might see informational messages similar to this:

```
Building dynamic ELF inventory file for rqs(1) processing .. 100% Done.
Invoking rqs(1) in necessary dynamic ELF objects .. 100% Done.
```

Stopping the Remote CD Manager

If your distribution source was a CD on a remote installation server, *cdinstmgr* is still running on the installation server (see “Preparing the Remote CD-ROM Drive” on page 26). Follow this procedure to stop *cdinstmgr*:

1. Go to the remote server or use a remote login.
2. Type “quit” at the prompt.

   This prompt is posted until you stop *cdinstmgr* (see “Preparing the Remote CD-ROM Drive” on page 26):

   ```
   CD on CDdir for host hostname: type the word "done" when
   you are finished with this CD, "quit" if you are
   completely done: quit
   ```

3. You might see this message from *cdinstmgr*:

   ```
   CD on CDdir for host hostname: the CD-ROM daemon was started
   by you,leave it running? (y/n)
   ```

   If you answer yes, the CD is not ejected and *cdromd*(1M), which was started by *cdinstmgr,* is not stopped. If you answer no, *cdinstmgr* attempts to eject the CD and stop *cdromd*. If the CD is in use, this message is displayed and the CD is ejected:

   ```
   CDdir: Device or resource busy
   ```

4. Eject the CD, if you wish.

   To eject the CD, issue the *eject*(1) command at the installation server or from the remote login session.

   ```
   % eject
   ```

   If the CD is in use, this command fails and the CD is not ejected.

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/passwd* 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/passwd.O*, for example). Then it installs the new, required version of the file by its standard name (*/etc/passwd*).
- 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*.

Note: 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 (which should subsequently be merged with the installed version and 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 an *.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 5-18, is repeated the next several times that the target system is rebooted.

```
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.
```

Example 5-18 Notice of Configuration File Changes

Merging Configuration Files

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

1. Identify changed configuration files with this command:

   ```
   # showfiles -cCH
   ```

   Example 5-19 illustrates output of the previous *showfiles* command.

   ```
   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
     /usr/etc/inetd.conf
   ? /usr/etc/inetd.conf.O
   ```

Example 5-19 Listing of Changed Configuration Files

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, since 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.

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 message 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.
2. Restore the previous installation session and let me look at it.
3. Ignore the previous installation session.
Please enter a choice [1]:
```

Figure 5-3 The Checkpoint Restart Menu

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

- Retry the previous installation (the default)

  If you select this action, Inst attempts to restore the previous session and continue the installation.

- Restore the previous installation session and let me look at it

  If you select this action, Inst displays the `Inst>` prompt. Use the *list i* command to see what remains to be installed, and then enter the *go* command.

- Ignore the previous installation session

  If you select this action, the previous session is aborted (pending selections are disregarded) and Inst is invoked 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 is exists or is unavailable. For example, the distribution source may be unaccessible because of a network failure or because the distribution was moved.

  If the distribution is missing, you see the following message when you select item 1 to restart the installation; when you select item 2, you see this message when you try to list the distribution:

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

- Different software distribution

  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, you see the following message when you select item 1 to restart the installation; when you select item 2, you see this message when you try to list the distribution:

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

- Bad selections

  This failure occurs when the installed software on the target or the product dependencies on the distribution have 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 this message:

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

  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 64 if you need instructions 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.

Part I
Advanced Installation Procedures

Chapter 6

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:

- "Quick Start Installation Instructions"
- "Alternative Ways to Specify a Distribution"
- "Installing Software Noninteractively"
- "Using Command Shortcuts"
- "Using step to Make Selections"
- "Using the View Commands Menu"
- "Using the Administrative Commands Menu"
- "Using the Interrupt Menu"
- "Giving Commands on Other Menus"

Quick Start Installation Instructions

Use this procedure if you are already familiar with using 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 you need to:

   ```
   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 51 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.

4. 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 54 for details.) If you omitted step 3, you see the initialization messages before the software inventory is posted.

5. 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 a display of subsystems that are installed on the target and their removal status. (See "Using the View Commands Menu" on page 92 for details.)

6. Step through the distribution inventory to change selections, as needed:

   ```
   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 59 and "Using step to Make Selections" on page 89 for more information.)

7. 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 62 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.

8. Resolve conflicts, if needed:

```
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 and Other Errors" on page 63 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.

9. 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.

10. 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 67 for details). Otherwise, return to step 3 to change the distribution source or to step 4 to display the distribution inventory.

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 5 for basic information). This section suggests several alternatives to the specification methods described in Chapter 5.

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 125). 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 issuing 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 19). 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 51 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 pre-selected by Inst (see “Required and Default Subsystems” on page 6 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.

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

Example 6-1 Sample of a Non-interactive Installation

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> 5
```

Abbreviating Command Names

Inst recognizes the shortest unique abbreviation of any command, so your command entry can be very brief. You can also use command abbreviations in combination with keywords (see “Using Keywords as Arguments” on page 86). 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 85 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
eoe1.sw.*	All *sw* subsystems in the eoe1 product
.man.	All *man* subsystems in the distribution
*.man	All *man* images in the distribution
eoe*	All products whose names begin with *eoe* (*eoe1* and *eoe2*, for example)
[a-c]*.sw.*	All *sw* 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 (.* and .*.*) 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.

```
Inst> list eoe2
Inst> install *.man
Inst> keep *
```

Example 6-2 Commands With Product and Image Arguments

You can always 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.

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

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

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 *showprods -a* output plus all subsystems in the software distribution.
default	d	All subsystems that are recommended for installation by the manufacturer. A *d* appears after the subsystem name in *list* output.

Table 6-2 Keywords

Keyword	Abbrev.	Meaning
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. A `D` appears in the second column of *list* output
install	i	All subsystems that are selected for installation. An `i` appears in the first column of *list* output
installable	A	All subsystems in the software distribution. When no keyword argument is given to the *list* and *step*, the *installable* keyword is used by default.
installed	I	All subsystems that have the same version installed and available for installation. An `I` appears in the second column of *list* output.
keep	k	All installed subsystems that are in the software distribution and are not selected for installation or removal.
miniroot	b	All subsystems that must be installed with a miniroot installation or that require a reboot to complete installation. A *b* appears after the subsystem name in *list* output.
newer	N or n	All subsystems in the distribution that are not and never were installed on the target. An `N` appears in the second column of *list* output.
notinstalled	(none)	All subsystems that were never installed on the target or were installed and later removed.
patches	(none)	All patches in the distribution.
patchupgrade	P	All patches on the distribution that are upgrades to installed patches. A *P* appears in the first column of *list* output.
removable	L	All installed subsystems that can be selected for removal.

Table 6-2 Keywords

Keyword	Abbrev.	Meaning
remove	r	All subsystems that are selected for removal. An *r* appears in the first column of *list* output.
removed	R	All subsystems that were once installed on the target and later removed. An *R* appears in the first column of *list* output.
required	q	All subsystems that are required for proper operation of the target. An *r* appears after the subsystem name in *list* output.
same	S or s	All subsystems for which the version on the distribution and the version in the target are the same. An *S* appears in the second column of *list* output.
target	targ	All subsystems on the target.
uninstalled	U	Subsystems that meet all of these conditions: they were available for installation in the past, they are not currently installed, they don't have a version in the distribution, and they are not *maint* subsystems.
upgrade	u	All subsystems for which the distribution contains a version that is an upgrade to the installed version on the target. A *U* appears in the second column of *list* 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 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 your current *view* settings. The *step* command also takes the same keyword arguments and wildcards as *list*.

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

```
Inst> step U
```

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

```
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 refind.man.rfind [d]            0 Rfind Man Pages
```

Example 6-4 A step Command Display

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 using 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 *d*.	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 *d*.	**`<Shift><right-arrow>`**
	Select the remaining subsystems in this product for removal.	**`R`**

Table 6-3 Requests in *step* Operations

Level Affected	Requested Action	Keys
	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 *d*.	**`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>`**
	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 *pattern* and display the product that contains it.	**`/`***pattern***`<Enter>`**
	Display help for *step* operation.	**`h`**
	Quit *step* operation.	**`q`**

Using the View Commands Menu

By default, the *list* and *step* commands display all software in the distribution at the subsystem level; subsystems are sorted alphabetically by shortname. Using the View Commands menu, you can change the default behavior of *list* and *step* to adjust their output. You can also use the *set* command from the View Commands menu to display and change Inst preferences.

To display the View Commands menu, enter the *view* command from the Main menu, as shown in Figure 6-1.

```
Inst> view
```

```
  View Commands Menu

   1. dist                         Show only products on distribution
   2. targ                         Show only products on target
   3. filter [setting]             Show or hide products by attribute(
   4. level [1,2,3]                View (1) product, (2) image or (3)
   5. sort [size,name]             Select sorting of product list
   6. show                         Show the current view settings
   7. return                       Go to the Inst Main Menu

  View>
```

Figure 6-1 The View Menu

To return to the Main menu from the View Command menu, issue the return command:

```
View> return
```

Viewing the Target and Distribution

The *dist* and *targ* item 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 location:

```
View> targ
Current Location : target
```

```
View> dist
Current Location : distribution
```

Any *list* or *step* command that you enter after changing the view contains software items from the specified location. Example 6-5 illustrates the effects on *list* output when the view is set to the target. Notice that the legend that appears above *list* output identifies the view location.

```
View> targ
View> return
Inst> list
Current View:
 Location: target
 Status: I=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
u I 4DDN.sw.4DDN [bd]            0 4DDN Software
u I 4Dwm.sw.4Dwm [d]             0 Desktop Window Manager
u I ViewKit_eoe.sw.base [d]      0 ViewKit Execution Environment, 1.1
```

Example 6-5 A list Display From the Target View

Example 6-6 illustrates effects on *step* output when you reset the view to the distribution.

```
View> dist
View> return
Inst> step
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.
4DDN.man.4DDN [d] 0 4DDN Manual Pages
```

Example 6-6 A step Display From the Distribution View

Using View Filters

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 off and on a specified subset of the display.

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 6-7 illustrates a target listing before and after the previous *filter* command is issued (the *list* legend is omitted from this example):

```
Inst> list

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

Inst> view filter removed off
Inst> list

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

Example 6-7 Effects of Using View Filters

Setting Viewing Levels

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 6-8 illustrates the effects of a *level* command on the first two products shown in Example 6-7.

```
View> level image
View> return
Inst> list
```

```
4DDN.man                0 4DDN Documentation
4DLT.sw                 0 4DLTsoftware3.1
InPerson.books          0 InPersonBooks
InPerson.man            0 InPersonDocumentation
InPerson.sw             0 InPersonSoftware
```

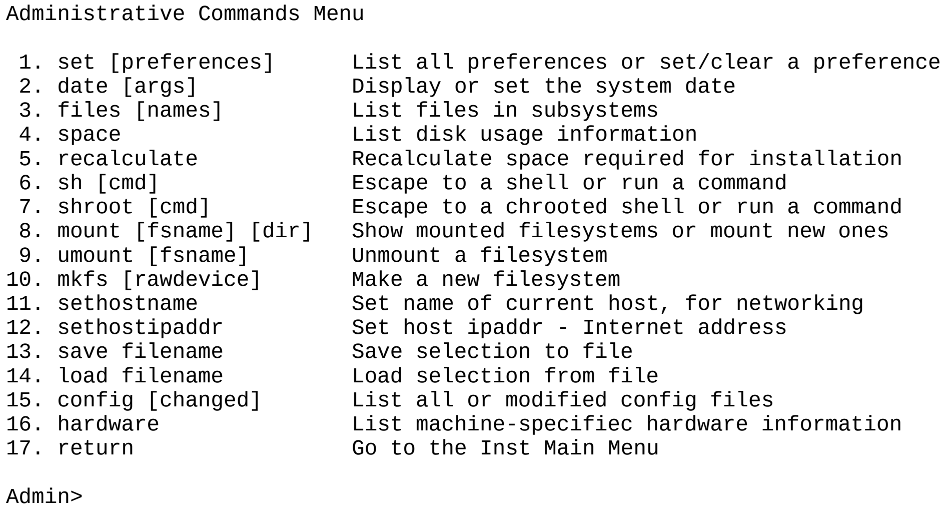
Example 6-8 Effects of level on a list Display

Using the Administrative Commands Menu

This section describes some commands from the Administrative Commands menu that may be required during a routine installation. For information on administrative commands that are not covered in this discussion, please refer to Chapter 9, “Inst Command Reference.”

Use this command from the Main menu to display the Administrative Commands 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. mount [fsname] [dir]   Show mounted filesystems or mount new ones
 9. umount [fsname]        Unmount a filesystem
10. mkfs [rawdevice]       Make a new filesystem
11. sethostname            Set name of current host, for networking
12. sethostipaddr          Set host ipaddr - Internet address
13. save filename          Save selection to file
14. load filename          Load selection from file
15. config [changed]       List all or modified config files
16. hardware               List machine-specifiec hardware information
17. return                 Go to the Inst Main Menu

Admin>
```

Figure 6-2 The Administrative Commands Menu

The most commonly used commands on the Administrative Commands menu are described in the subsections that follow (an exception is the *set* command, which is the subject of Chapter 8). For information on the remaining commands on this menu, please refer to Chapter 9, "Inst Command Reference," or the online help that is available in Inst.

Displaying the Files in a Subsystem

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

Example 6-9 shows the format and output of the *files* command.

```
Admin> files eoe1.sw.quotas
```

```
i U  eoe1.sw.quotas        1  f etc/config/quotacheck
i U  eoe1.sw.quotas        1  f etc/config/quotas
i U  eoe1.sw.quotas           f etc/init.d/quotas
i U  eoe1.sw.quotas        1  l etc/rc2.d/S10quotas
i U  eoe1.sw.quotas      193  f usr/bsd/quota
i U  eoe1.sw.quotas      145  f usr/etc/edquota
i U  eoe1.sw.quotas      137  f usr/etc/quot
i U  eoe1.sw.quotas      145  f usr/etc/quotacheck
i U  eoe1.sw.quotas        1  l usr/etc/quotaoff
i U  eoe1.sw.quotas       25  f usr/etc/quotaon
i U  eoe1.sw.quotas      129  f usr/etc/repquota
i U  eoe1.sw.quotas       28  f usr/sysgen/boot/quotas.o
```

Example 6-9 Sample files Display

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 an `i` or is blank) or removed (the first column contains an `r`). The *files* list does not include the components of a currently installed subsystem that are not included in the new software, since 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:

f	Plain file
d	Directory
b	Block special file
c	Character special file
l	Symbolic link
p	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 issue this command:

```
Amin> files names
```

Escaping to the Shell

While you are using Inst, it is possible to invoke a shell, enter shell commands, and return to Inst. 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 the value of the SHELL environment variable, which is usually */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*. So 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*. Since the miniroot supports only a small number 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.

Making New Filesystems

During a miniroot installation, you can use the *mkfs*(1M) command to create filesystems. This command erases the entire contents of filesystems, so it should be used with caution. You are prompted through the *mkfs* command execution and required to confirm your command before *mkfs* is completed.

If the *mkfs* command is entered with no arguments, the root and user filesystems (*/* and */usr*) are erased and recreated. In addition to erasing the contents of */* and */usr*, recreating the root and user filesystems 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.

To make other filesystems, give device name arguments to *mkfs*:

```
Admin> mkfs device
```

See the *mkfs*(1M) manual page for more information about the device argument. To access the newly created filesystem, *device*, you must create a directory mount point and mount the filesystem:

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

Mounting and Unmounting Filesystems

You can identify mounted filesystems by entering the *mount*(1M) 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 */*.

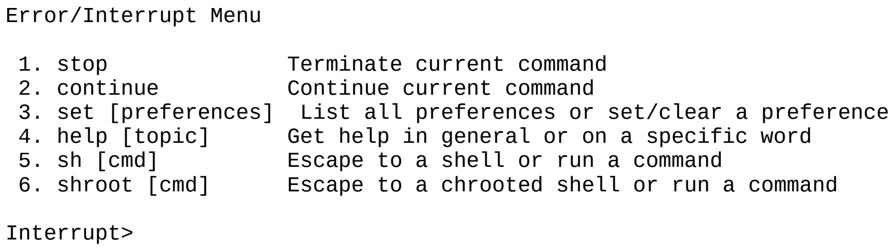
To unmount filesystems, you must be performing a miniroot installation. The *umount*(1M) command has this syntax:

```
Admin> umount filesystem
```

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. Also, under certain error conditions, the Interrupt Menu is automatically invoked by Inst. The Interrupt menu contains selections to cancel the command in progress, change preferences, open an IRIX shell, and get help.

Figure 6-3 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 preference
 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 6-3 The Interrupt Menu

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 remaining selections that were pending.

Caution: Using *stop* during installation can leave software in an inconsistent state that requires a 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 125 for information on using the *set* command; see Table 9-1 for a list of topics covered by the *help* command; and see "Escaping to the Shell" on page 98 and "Interrupt Menu Commands" in Chapter 9 for details on using the *sh* and *shroot* commands.

Giving Commands on Other Menus

From the Inst Main menu, you can give commands on the Administrative Commands menu without switching menus. To do this, enter *admin* and the command that you want at the `Inst>` prompt. For example:

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

You can also give Inst Main menu commands from the Administrative Commands menu without switching menus. To do this, enter **`return`** and the command that you want at the `Admin>` prompt. For example:

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

Chapter 7

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 a Software Update"
- "Installing Software Options"
- "Installing Patch Releases"
- "Installing Software for Hardware Upgrades"
- "Installing Reference Pages"
- "Reinstalling an Older Software Release"
- "Avoiding Compatibility Problems"
- "Removing All Software"

Installing a Software Update

When you receive a software update, it might be delivered on multiple CDs. Use the sequence numbers on the CD labels or the directions in "Planning the Order of Installation" on page 23 to determine the order in which you should install the updates. Updates are not necessarily included for all products, since not all products are updated at the same time.

When you install a software update, replace or remove all, not just some, of the older subsystems in each product that you install. When a server or workstation contains software subsystems from different releases of a product, compatibility problems result that can be very difficult to diagnose. Use *showprods* with the *–n* argument to detect subsystems that are not from the same product release.

When Inst makes its initial installation selections, it automatically selects any updates to subsystems that are already installed. In addition, it selects subsystems that are new with this release and designated as default ("Required and Default Subsystems" on page 6). Subsystems that you omitted from the previous installation, even subsystems with a default designation (*d* in *list* output) are not selected for installation. In many cases, you can examine the selections with *list* and give the *go* command without making any selection changes.

If you see a message about changed configuration files after the installation is complete, follow the directions in "Merging Configuration Files" on page 71 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 Software Options

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:

- *T*he release notes for a software option explain whether a miniroot installation is required.
- If the software option requires miniroot installation, use the installation tools that accompanied the version of *eoe1* that is already installed on your system.
- If you have several software options to install and they are on more than one CD or distribution directory, use the CD sequence numbers, release notes information, or information in "Planning the Order of Installation" in Chapter 2 to determine the installation order. If the installation order is wrong, Inst will advise you when you launch the installation.

Installing Patch Releases

A patch release contains replacement files that can be installed to modify a particular software product. Installing patches is optional; review the online release notes that accompanied the patch to determine whether to install it or not.

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.

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.

Patch Installation Procedure

Use this procedure as a guideline to installing a patch release:

1. 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; otherwise, invoke Inst from the IRIX command line.

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

2. 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 distribution contains several patches, select the subsystems in any additional patches that you want to install.

   ```
   Inst> install patchname
   ```

3. Issue the *go* command to launch the installation.

4. Resolve any conflicts.

   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.

5. 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, invoke Inst from the miniroot to remove the patch (the miniroot is required to reinstall the original product files, in the case); otherwise, invoke Inst from the IRIX command line.

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. Issue 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*(1M) 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*(1M) 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 for the upgrade.

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 Software Options” on page 104 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 reinstalling software after a hardware upgrade, be sure to follow the directions in "Planning the Order of Installation" on page 23 to determine installation order. Reinstallations require a miniroot installation.

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.

   Since 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 I
   ```

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 what reference page subsystems are installed:

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

- List all reference page subsystems available for installation:

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

- Select all reference page subsystems, but 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 *eoe1* and *eoe2* 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 diagnosing compatibility problems very difficult.

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, not IRIX.

Before beginning this procedure you should 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 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 satisified 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 enty.

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 finsihed 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 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 6). 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 your 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 71'. 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 of 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. The prevents incompatibilities that can occur when you switch from one installation session to another.

Removing All Software

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

Chapter 8

Tailoring Inst Operation

Various features of Inst operation can be tailored using the *set* command on the Administrative Commands menu. Using *set*, you can assign values to arguments known as *preferences*, which control aspects of the Inst interface. This chapter explains how to use *set* to adjust menus and information displays, control space calculations, and modify other Inst operating defaults. A table of available *set* preferences is included at the end of the chapter (a complete list is also available from Inst online help).

This chapter contains the following sections:

- "Checking the Current Settings"
- "Controlling Menu Displays"
- "Controlling Disk Space Checks and Reports"
- "Controlling the Display of Filenames"
- "Displaying the Current Distribution Source"
- "Turning Off Initial Selection of Subsystems"
- "Changing the Network Timeout Value"
- "A Closer Look at Preferences"

Note: The procedures in this chapter assume that *set* commands are entered at the `Admin>` prompt; however, except where noted, *set* commands can also be entered at the `Inst>` and View> prompts.

Checking the Current 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, give 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 is saved in the installation history database; its value is maintained for subsequent Inst sessions.

Redisplaying the Menu

Since 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 pseudo-command 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 the / and */usr* filesystems (and any other mounted filesystems), the disk space required for each subsystem, and also 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 Detail 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 issue 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 Detail 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 detail space checking, issue this *set* command before you issue 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 detail 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, issue 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 delaying 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, issue 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, issue 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 82).

Forcing Disk Space Recalculation

If you modify files on the target system while working in an IRIX shell (using *sh* or *shroot*), Inst automatically updates disk space calculations. You can also issue the *recalculate* command from the Administrative Commands menu to request a new space check:

```
Admin> recalculate
```

This command always performs a detail space check, even when the *detailspacecheck* preference is off.

Displaying the Disk Space Summary

Disk space summaries are displayed at the end of the output from *list* and *step* commands. You can also display 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):
                                 /
Selections net change        2853
Space available            547421
```

The *space* command does not force a recalculation of disk space, so if you issue the *space* command before space calculation are performed, the `net change` amount is shown as 0 blocks, even if you have made installation or removal selections.

Displaying Absolute Sizes

By default, the *list* command displays relative disk space information: the increase or decrease in disk usage that occurs if a subsystem is installed or removed. The amount is displayed in kilobytes and is followed by a plus sign (when additional space is required) or minus sign (when space is freed). Using the *show_absolute_sizes* preference, you can set *list* to show absolute rather than relative size information.

This example illustrates how to display absolute subsystem sizes:

```
Admin> set show_absolute_sizes on
Admin> return
Inst> list sizes motif_dev
...
    Subsystem                              Description

    motif_dev.man.dev *            2377  IRIX IM Dev Manual Pages
    motif_dev.man.relnotes *         17  IRIX IM Technical Notes
    motif_dev.sw.demo              3721  IRIX IM Demos & Examples
    motif_dev.sw.dev *             7037  IRIX IM Libraries & Headers
...
Inst>
```

Example 8-1 Sample Display of Absolute Sizes

The *delayspacecheck* preference must be off to display absolute sizes in *list* output.

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 file names are displayed.

To display file names as they are installed, give this command before you enter the *go* command:

```
Admin> set show_files on
```

To return to no display of filenames as they are installed, give 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, give this command:

```
Admin> set overprint off
```

To return to the default behavior, give 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 given in “Step 2: Specifying the Source” on page 51 and “Alternative Ways to Specify a Distribution” on page 82.

Each time you issue the *from* command, the current distribution source changes. You can display the current distribution source by issuing 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 issue 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 selections are still erased.

To prevent Inst from making selections after you issue the *from* command, give 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 yourself. 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 off, it remains off in future Inst sessions until you change its value.

Checking and Setting the System Date

The correct date and time of installations is important since portions of the installation process use *make(1)*, which relies on dates to determine whether certain commands should be given. The correct date and time ensure that the actions that are sensitive to the dates of files are performed correctly.

The *date* command on the Administrative Command menu is identical to *date(1)*. This command prints the date:

```
Admin> date
```

This command sets the date:

```
Admin> date mmddhhmmyy
```

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

Changing the Network Timeout Value

Inst allows you to specify a network timeout 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 timeout 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 timeout 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).

Some preferences, known as *permanent* preferences, retain their settings until they are reset. Other preferences, known as *transient* preferences, revert to a default setting at the start of a new session. A group of preferences have 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 List

Table 8-1 describes the preferences that you can specify with the *set* command (or from the Customize Installation menus in Software Manager). It lists the name of the preference, describes the function that the preference performs, and lists 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 Preferences and Their Functions

Preference Name	**Function**	**Default Setting**	**Value Saved?**	**Comments**
abort_on_error (Expert)	Controls whether an automatic installation is aborted if an error occurs. If *true* (default), an error cancels the installation; if *false*, the installation continues after the error occurs. Regardless of setting, errors are reported in */var/inst/INSTLOG*.	true	yes	N/A
always_confirm_quit	Controls whether the user is always asked to confirm a *quit* command. If *true*, user must confirm a quit; if *false* (default), no user confirmation is needed after *quit*.	false	yes	N/A
always_page_inst	Controls whether requested information, such as progress messages during *go* processing, are displayed in page segments or scrolled off the screen. If *true*, requested information is displayed in page segments; if *false* (default), information that is longer than a page scrolls off the screen.	false	no	Applies only to Inst.
auto_inst_new	Controls whether new products are selected for installation when installation is automatic. If *true*, all new products are installed during an automatic installation. If *false* (default), new products are omitted from automatic installations.	false	yes	Applies only to Software Manager.
auto_inst_upgrades	Controls whether upgrade products are selected for installation when installation is automatic. If *true* (default), all upgrade products are installed during an automatic installation. If *false*, upgrade products are omitted from automatic installations.	true	yes	Applies only to Software Manager.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
autoselect	Controls whether *inst* does an automatic selection of subsystems during initialization and when a new distribution source is specified. If *true* (default), *inst* automatically selects subsystems at these events in the session; if *false*, no subsystems are automatically selected.	true	yes	N/A
background (Expert)	Controls whether Software Manager is run as a background or foreground task when invoked from an IRIX shell. If *true* (default), Software Manager runs as a background task; if *false*, Software Manager runs as a foreground task.	true	yes	Applies only to Software Manager.
beep (Expert)	Controls whether an audible signal is given *after go* processing is complete. If *true* (default), audible signal rings when *go* processing finishes; if *false*, no signal is given	true	yes	N/A
busy_check (Expert)	Controls whether files that are currently in use by active processes are included in the pre-installation space calculations. If *true* (default), the amount of space required by busy files is included in calculations; if *false*, busy files are omitted from calculations.	true	no	Always false for miniroot installations.
checkpoint_restart (Expert)	Set during a session recovery, when the user selects the action to take after an abnormal exit. If *true*, the user chose to retry the interrupted installation using the selections made during the session; if *false* (default), the user chose to either return to the previous session without attempting to install selected software or to ignore the previous session completely.	false	no	N/A
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 *true*, user selections are saved in the file as soon as they are made; if *false* (default), pending selections are saved when the user enters the *go* command.	false	yes	N/A

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
clearprompt	Controls how throw-away prompts are displayed. If *true* (default), these prompts are cleared from a line by overwriting with blank spaces; if *false*, the original prompts remain on the line and a new line is generated to display a new prompt.	true	yes	Applies only to Inst; set to false on terminals that do not perform line clearing.
confirm_quit	Controls the confirmation prompt for *quit* commands when selections are pending. If *true* (default), users are notified if install or removal selections are pending when they enter *quit*; if *false*, users are allowed to quit the session without a reminder of pending selections.	true	yes	N/A
confirm_nfs_installs	Controls whether user is prompted to confirm installation on NFS mounted filesystems. If *on* (default), user is prompted for confirmation; if *off*, files are installed in the NFS filesystem if the user has proper permissions.	on	yes	N/A
custom_startup_mode (Expert)	Controls how Software Manager is initially presented. If *off* (default), Software Manager always starts in Automatic Installation mode (selections are automatically made at initialization). If *distribution*, 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 *always*, 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	Applies only to Software Manager.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	**Function**	**Default Setting**	**Value Saved?**	**Comments**
default_config (Expert)	Controls handling of configuration files. If *true*, all configuration files that are in the distribution are installed; if the target version was modified, it is saved in a file with a *.O* extension. If *false* (default) and the target version was modified, install the distribution version of the configuration files, appending a *.N* extension; if the target version was not modified, overwrite it with the new file.	false	yes	N/A
default_sharedirs (Expert)	Controls the location of directories that are shared by diskless clients. The default location is */usr*.	/usr	no	N/A
delayspacecheck	Controls the timing of disk space calculations. If *on*, disk space calculations are deferred until the *go* or *space* commands are issued, and no space information is provided when *list* or *step* commands are entered; if *off* (default), space calculations are performed at the first *list*, *step*, or *space* command, unless disk space checking was done previously.	off	yes	N/A
detailspacecheck	Controls the source of the data used in space checking. If *on*, space requirements for files is determined by checking their sizes on the disk; if *off* (default), file sizes are assumed to be the value that is recorded in the installation history database.	off	yes	Set *on* for slower, more accurate, calculations.
display_legend	Controls the legend for *list* displays. If *true*, the output of *list* contains an explanation of the mnemonic tags that appear in the listing; if *false*, the legend is not posted.	true	yes	Applies only to Inst.
display_size	Controls the unit of measure for subsystem sizes. Valid settings are *kbytes* (default), *kilobytes*, *bytes*, or *512-byte blocks*.	kbytes	no	Applies only to Inst.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
dist	Specifies the value of the current distribution source. For live installations, the default value is the previously specified source. For miniroot installations, *dist* must be specified.	source	yes	Product names are not saved as part of the value.
error.display_verbosity (Expert)	Controls the level of detail in errors that are displayed on the screen. If *2*, the display contains the maximum level of detail; if *1* (default), the display contains moderate level of detail; if *0*, no details are displayed.	1	yes	N/A
error.log_verbosity (Expert)	Controls the level of detail in errors that are recorded in the log file. If *2* (default), the log contains the maximum level of detail; if *1*, the log contains moderate level of detail; if *0*, no details are logged.	2	yes	N/A
fullmenu	Controls the display of hidden commands (but not their availability) on the Main menu, Administrative Commands menu, and View Commands menu. If *on*, hidden commands appear on these menus; if *off* (default), hidden commands do not appear.	off	yes	Applies only to Inst.
hide_image_products (Expert)	Controls whether the image components of each product are displayed when the product list is displayed. If *true* (default), listings of products do not include their component images. If *false*, the image components of a product are displayed in product listings.	true	yes	N/A
info.display_verbosity (Expert)	Controls the level of detail in informational messages that are displayed on the screen. If *2*, screen messages contain the maximum level of detail; if *1*, screen messages contain a moderate level of detail; if *0*, screen messages contain no details.	0	yes	N/A

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
info.log_verbosity (Expert)	Controls the level of detail in informational messages that are recorded in the log file. If *2*, logged messages contain the maximum level of detail; if *1* (default), logged messages contain a moderate level of detail; if *0*, logged messages contain no details.	1	yes	N/A
inst_initial_level (Expert)	Controls the initial setting of the *view* command. If *subsystem* (default), the output of *list* displays the subsystems in each product that appears in the listing; if *image*, the output of *list* displays the images for each product that appears in the listing; if *product*, the output of *list* displays product names without product components.	subsystem	yes	N/A
inst_visible_resources	Controls the list of preferences that are displayed when the *set* command is issued with no arguments (see also *swmgr_visible_resources*). This preference takes multiple values: *tty* (default) specifies that the displayed list should contain all preferences that apply to the Inst interface; *permanent* (default) specifies that the displayed list should contain all preferences whose value is maintained in subsequent sessions (until the value is reset); *transient* specifies that the displayed list should contain all whose value reverts to a default at the end of a session; *beginner* (default) limits the displayed list to all preferences that are not designated *expert*; *expert* specifies that the displayed list contain all preferences whose settings have potentially adverse effects on installation.	tty permanent transient beginner	yes	Applies only to Inst. In your entry, separate values with a comma (no spaces are allowed).
install_identical_files (Expert)	Controls whether files are installed if the distribution version appears identical to the version installed on the target. If *true*, identical distribution files are installed. If *false*, distribution files are not installed when they appear identical.	true	yes	Set to *false* for faster installations.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
install_sites (Expert)	Contains the list of previous distribution sources (also see the *install_site_size* preference).	N/A	yes	N/A
install_site_size	Specifies the number of items that are displayed in the list of previous distribution sources. Valid settings are integers.	10	yes	N/A
menus	Controls whether menus are displayed automatically. If *on* (default), Main menu is displayed at startup and all menus are displayed when they are invoked; if *off*, menus are not displayed at startup or invocation, but users can display them manually by entering *?* at the menu prompt.	on	yes	Applies only to Inst; set *off* for experienced Inst users.
network_retry	Specifies the number of network timeouts that will be allowed before connection attempts are abandoned.	2	yes	N/A
never_resize_pane (Expert)	Controls whether pane is resized when switching the selection mode (see *custom_startup_mode* preference). If *true* (default), the pane size remains the same for all selection modes. If *false*, pane size is changed when the selection modes are changed.	true	yes	Applies only to Software Manager.
neweroverride	Controls whether older products can replace newer ones. If *on*, users can install older products in place of new ones; if *off* (default), older products cannot be installed when a more recent version exists on the target.	off	no	N/A
overprint	Controls the display of the software listing during installation (or removed). If *on* (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 *off*, a new line is used to display each name; the list scrolls down the screen.	on	yes	Applies only to Inst.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	**Function**	**Default Setting**	**Value Saved?**	**Comments**
override_space_check (Expert)	Controls whether space checking is performed. If *true*, no space checking is done during an installation, allowing disk overflows. If *false* (default), space checking is performed.	false	no	Setting *true* can cause serious installation problems.
page_output	Controls tty device output. If *true* (default), output is managed by paging that is similar to more(1M); if *false*, output scrolls off the screen.	true	no	Applies only to Inst.
post_install_dialog (Expert)	Controls whether a completion notice is posted when an installation is completed. If *true* (default), a completion notice is posted. If *false*, the user sees a message only if a reboot is required.	true	yes	N/A
precious_files (Expert)	Specifies the list of files that cannot be overwritten or removed during the installation. Value is one or more filenames that are specified as arguments.	None	no	In your entry, separate values with a comma (no spaces are allowed).
promptforid (Expert)	Controls whether user is prompted to specify unknown user IDs (UIDs) and group IDs (GIDs). If *true*, user is prompted for a UID if one is not found in */etc/passwd* and a GID if one is not found in */etc/group*. If *false* (default), inst assigns a UID and GID based on the *startid* preference and writes them to */usr/adm/SYSLOG*.	false	yes	N/A
rulesoverride	Controls whether conflicts can be overwritten. If *on*, incompatible subsystems and subsystems without prerequisites may be installed on the target. If *off* (default), incompatibilities and prerequisites must be resolved before installation is done.	off	no	Use *on* setting with caution.
selections_file (Expert)	Specifies the name of the file that contains software selections during the session.	N/A	no	N/A

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
set_path (Expert)	Specifies the value to use when setting the *$PATH* variable for exit operations. If *default* (default), set to a known path; if *environment*, set path to *$PATH* environment variable or use the default path if *$PATH* is not set; if *specific*, use the value in *set_path_specific* preference.	default	yes	N/A
set_path_specific (Expert)	Specifies the path to use for exit operations. Value is a user-supplied string.	N/A	yes	N/A
shell	Specifies the program to use for the *sh* and *shroot* Administrative commands. Usually this value is either */bin/sh* or */bin/csh*.	/bin/csh	yes	N/A
short_names	Controls product names in listings. If *true*, only the shortname is displayed in listings; if false, product descriptions are displayed instead of their shortnames.	false	yes	N/A
show_absolute_sizes	Sets *list* and *step* 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 *on*, the absolute subsystem size is displayed; if *off* (default) the relative size is displayed.	off	no	Default unit of measure is kilobytes. See also *display_size*.
show_files (Expert)	Controls whether filenames are displayed during installation. If *true*, each component file in a product is displayed as it is installed; if *false* (default), the product name is displayed as it is installed.	false	yes	N/A
show_lint (Expert)	Determines whether inconsistencies in the distribution are displayed during reading (for debugging). If *true*, inconsistencies are displayed; if *false* (default), inconsistencies are not reported.	false	yes	N/A

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
space_check (Expert)	Controls whether space checking is performed. If *true* (default), space checking is done at various points in the installation (see the *delayspacecheck* preference). If *false*, no space checking is performed.	true	no	Setting *false* can cause serious installation problems.
space_indicator	Controls format of space-checking information. Valid settings are *pie*, *bar*, and *text*.	pie	yes	Applies only to Software Manager.
startid	Sets the starting value of UIDs and GIDs when this information is unknown to *inst* and *promptforid* preference is set off (also see *promptforid* preference). The value of *startid* is assigned to the first unknown UID and GID; subsequent unknown UIDs and GIDs are incrementa by 1.	6000	yes	N/A
swmgr_initial_level (Expert)	Controls the level of the product hierarchy that is displayed when Software Manager is invoked. Valid settings are *product* (default) and *subsystem*.	product	yes	N/A
swmgr_visible_resources	Controls the set of preferences that are visible in the Preferences dialog (see also *inst_visible_resources*). These settings are valid: if *none*, no preferences are displayed in the dialog; if *permanent* (default), the dialog contains all preferences whose values are retained for subsequent installation sessions (until the value is changed); if *transient*, the dialog contains preferences whose settings revert to a default at the end of a session; if *beginner* (default), the dialog contains all preferences that are not designated *expert*; if *expert*, the dialog contains preferences whose settings have potentially adverse effects on installation.	beginner permanent	yes	Applies only to Software Manager. In your entry, separate values with a comma (no spaces are allowed).
timeout	Specifies the amount of time, in seconds, that *inst* allows for a reply from a remote host before retrying the connection. Two retries are done after the initial attempt.	120 seconds	yes	Set to a value that is suitable for your network.

Table 8-1 *(continued)* Preferences and Their Functions

Preference Name	Function	Default Setting	Value Saved?	Comments
use_last_dist (Expert)	Controls the contents of the Available Software field. If *true*, the last distribution source that was used is automatically entered in this field at startup. If *false*, this field is empty at startup.	false	yes	Applies only to Software Manager.
verify_checksum (Expert)	Controls whether files are verified by checksum as they are uncompressed or read from the archives. If *true*, checksum verification is performed. If *false* (default), no checksum verification is made.	false	yes	N/A
warning.display_verbosity (Expert)	Controls the level of detail in warning messages that are displayed. If *2*, warning messages contain the maximum level of detail; if *1*, warning messages contain a moderate level of detail; if *0* (default), warning messages contain no details.	0	yes	N/A
warning.log_verbosity (Expert)	Controls the level of detail in warning messages that are recorded in the log file. If *2*, logged messages contain the maximum level of detail; if *1* (default), logged messages contain a moderate level of detail; if *0*, logged messages contain no details.	1	yes	N/A
wrapmode	Controls whether displayed lines will wrap or truncate at the screen width limit. If *wrap* (default), lines longer than the screen width wrap to the next line; if *truncate*, lines are truncated at the limit of the screen width.	wrap	yes	Applies only to Inst

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.*

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.

You or 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 *.swmgrr*, 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 Format of a Preferences File

Each line of the *.swmgrrc* file contains the name of a preference and its assigned values. Example 8-2 illustrates the format of *.swmgrrc* entries. It illustrates settings that are expressed as boolean, single choice, multi-value, and integer values.

```
auto__inst_new: true
display_size: blocks
inst_visible_resources: tty,transient,permanent,beginner,expert
network_retries: 4
```

Example 8-2 Sample *.swmgrrc* File

Part II
Inst Command Reference

Chapter 9

Inst Command Reference

This chapter provides a description of the commands listed on the primary menus of the Inst application. It describes commands that are not covered in other parts of this guide, including hidden commands, and provides expanded discussions of command features. Commands on each menu are presented in alphabetical order.

This chapter contains these sections:

- "Main Menu Commands"
- "View Menu Commands"
- "Administrative Menu Commands"
- "Interrupt Menu Commands"

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.

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 82.

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 125 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 125 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 issue 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 after one screenful. Press the space bar when you are ready to display more output.

A complete list of *topic* arguments to *help* is shown in Table 9-1.

Table 9-1 Help Topics

*	devices	main	same
.N	dir	maintenance	save
.O	dist	miniroot	selection
?	distribution	mkfs	set
A	files	more	sethostipaddr
D	filter	mount	sethostname
I	from	names	sh
L	fsname	networkerror	shroot

Table 9-1 Help Topics

N
O
S
T
U
admin
all
args
b
base
beginner
check
config
conflicts
d
date
default
go
hardware
help
i
image
incompatible
inst
install
installable
installed
k
keep
kernelerror
keywords
list
load
lverror
new
nfs
older
options
overview
preferences
prerequisite
product
quit
r
readerror
recalculate
removable
remove
request
required
return
source
space
spaceerror
step
subsystem
targ
target
timeouterror
umount
uninstalled
unmount
upgrade
view
wildcard

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 issue 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 56 for a description of the output. Also see “Using the View Commands Menu” on page 92 for information on controlling *list* output.

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 125.

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 "Escaping to the Shell" on page 98 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 "Escaping to the Shell" on page 98 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 89 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 9-2 describes the effects of keystrokes during *step* operations. The items in this table assume that the viewing level is subsystem (see "Using the View Commands Menu" on page 92 for information on setting viewing levels).

Table 9-2 Keys Used in *step* Mode

Key	Action
d	Select this subsystem for installation if marked with *d*
f	List the files in this subsystem

Table 9-2 Keys Used in *step* Mode

Key	Action
h	Display help for the *step* 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 *d*
`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
U	(Same as **`I`**)

Table 9-2 Keys Used in *step* Mode

Key	Action
<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 92 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 125 for details).

View Menu Commands

dist

Use the *dist* command to set the view to the distribution. When the view is set to *dist*, the *list* and *step* commands display software in the distribution inventory. See also the *targ* command description in this section.

filter

Use the *filter* command with no arguments to display the list of filters that can be used to modify the output of *list* and *step* 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 *filter* command with one of these arguments to modify the output of *list* and *step* commands:

```
new
upgrade
patchupgrade
same
downgrade
notinstalled
```

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 9-1.

level

Use the *level* command with no arguments to display the current level of the software hierarchy (see Figure 1-1) 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 issue the *quit* command and are restored the next time that you invoke Inst. “The Preferences List” on page 125 describes *set* preferences in detail.

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 obsoleted 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*(1) or *gdiff*(1) 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.

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*(1) or *gdiff*(1) to compare the two versions of the files and add the *.N* version changes to the no-suffix version if you want them.

date

Use the *date* command with no arguments to display the system date. Enter arguments described in *date*(1) 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 "Displaying the Files in a Subsystem" on page 96 for more information.

hardware

Use the *hardware* command to display information about the internal architecture and components of the target system.

help

Use the *help* command to display Inst online help. Help is available on all topics listed in Table 9-1.

load

Use the *load* command to open a file that was created with the *save* command to contain a custom selection of software. Also see the discussion of *save*.

mkfs

Use the *mkfs* command to create a new filesystem (replaces the *clean* command found in earlier versions of Inst). You can use *mkfs* in miniroot installations only.

If a disk is new or if the current contents of the disk or filesystem are to be completely removed (cleaned), you can use *mkfs* 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.

When you use *mkfs*, specify the name of the disk device to be used for the filesystem. For example, the following command

```
mkfs /dev/dsk/dks0d1s6
```

creates an empty filesystem on partition 6.

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.

mount

Use the *mount* command without arguments to display mounted filesystems. Use *mount* 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 *mount* command with arguments can be used during miniroot installations only.

recalculate

Use the *recalculate* command to initiate a disk space calculation. The *recalculate* 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.

If the *detailspacecheck* preference is off, or if files were created or removed in another shell, use *recalculate* to resynchronize space information.

return

The *return* command returns 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
```

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 (also see 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 issue the *quit* command, and are restored the next time that you invoke Inst. "The Preferences List" on page 125 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 "Escaping to the Shell" on page 98 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 "Escaping to the Shell" on page 98 for additional information).

space

Use the *space* command to display a disk space summary; the displayed summary is the same as the *list* command if the *set* preference *delayspacecheck* is *off*. If *delayspacecheck* is *on*, the *space* command forces the disk space calculation to be performed and displays the disk space summary.

sethostname Use the *sethostname* command to the hostname of the target system; this command restarts networking using the new hostname. The *sethostname* command can be used in miniroot installations only.

sethostipaddr Use the *sethostipaddr* command to set the IP address of the target system; this command restarts networking using the new IP address. The *sethostipaddr* command can be used in miniroot installations only.

umount Use the *umount* command to unmount disks that Inst has mounted by default or that have been mounted with the *mount* 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.

Interrupt Menu Commands

The list below describes all commands that appear on the Interrupt menu.

continue Use *continue* to resume processing from the point at which Inst was interrupted.

help Use the *help* command to display Inst online help. Help is available on all topics listed in Table 9-1.

retry Appears only if Inst displays the Interrupt menu automatically, which happens if an error occurs during *go* processing (subsystems are being installed or removed). Use *retry* 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 issue the *quit* command and are restored the next time that you invoke Inst. “The Preferences List” on page 125 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 "Escaping to the Shell" on page 98 for additional information).

shroot Use the *shroot* command with no arguments to invoke a "chrooted" IRIX shell (see *chroot*(1M)). 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 "Escaping to the Shell" on page 98 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. So, Inst will 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 remaining 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.

Part III
Troubleshooting Installations

Appendix A

Installation Troubleshooting

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 or instructions are provided to help in correcting 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

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 which 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.

Getting More Feedback from Inst

The preferences described in Table A-1 control the amount of information that Inst provides in messages and warnings.

Table A-1 Preferences for Controlling Inst Feedback

Preference	Meaning
info.display_verbosity	A higher value results in more status messages, such as the name of each subsystem during an installation.
warning.display_verbosity	A higher value results in more warning messages, which are normally hidden. Increasing this value may assist in resolving other installation problems.
error.display_verbosity	Controls the level of detail in errors that are displayed on the screen. If *2*, the display contains the maximum level of detail; if *1* (default), the display contains a moderate level of detail; if *0*, no details are displayed.
error.log_verbosity	Controls the level of detail in errors that are recorded in the log file. If *2* (default), the log contains the maximum level of detail; if *1*, the log contains a moderate level of detail; if *0*, no details are logged.
show_files	Print the name of each file as it is installed or removed.
overprint	If *show_files* is on, then filenames are printed on the same line, using a backspacing effect similar to terminal device displays.

Refer to “A Closer Look at Preferences” on page 125 for more information on preference settings.

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 is text describing the possible causes and solutions to the condition which 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 162
- “Errors While Starting an Installation Session” on page 171
- “Pre-Installation Check Errors” on page 176
- “Errors While Installing and Removing Software” on page 178
- “Errors On Leaving an Installation Session (RQS Errors)” on page 183

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 163
- “Errors Loading the Miniroot From Remote Host” on page 165
- “Other Errors Loading the Miniroot” on page 169

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`**.
- No CD is in the CD-ROM caddy.
- No caddy is 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. Afterwards, the older PROMs do 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 196.

```
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 doesn't 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 196.

```
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 196," 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 Installation” on page 194 for a discussion of network problems that may occur during a miniroot installation.

```
No server for server:path(sash.cpu)
Unable to load bootp()server:path(sash.cpu):file not found
```

or

```
No server for server:CDdir/dist(sash.cpu)
open(bootp)server:CDdir/dist(sash.cpu) failed, errno = 6
Unable to load bootp()server:CDdir/dist(sash.cpu):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>(sash.cpu):
''bootp()server:path/sa(sash.cpu)'' is not a valid file to boot.
```

or

```
TFTP error: Access violation (code 2)
bootp()server:path/sa(sash.cpu): invalid
Unable to load bootp()server:path/sa(sash.cpu):
''bootp()server:path/sa(sash.cpu)'' is not a valid file to boot.
```

or

```
bootp()server:path/sa(sash.cpu): invalid
Unable to load bootp()server:path/sa(sash.cpu):
''bootp()server:path/sa(sash.cpu)'' 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 */usr/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 190).
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 189).

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 189).
10. Check the distribution directory as described in “Checking Distribution Directories and CDs” on page 199.
11. Perform any additional procedures that are necessary to identify the problem. See the section “Resolving Network Problems” on page 189.

```
file file not found in server:path/sa; directory contains:
...
Unable to load bootp()...
```

or

```
File CDdir/dist/sa not found on server server
Unable to load bootp()server:CDdir/dist/sa(sash.cpu): 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 `sash`.*cpu*, `sash`*cpu*, or `sashARCS` 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 you entered and look for a spelling or capitalization error, the wrong CPU in the `sash.`*cpu* portion of the command, `sash.IP12` rather than `sashIP12`, `sash.IP17` rather than `sashIP17`, or any *sash* command with `IP19`, `IP20`, or `IP22` rather than `sashARCS`. 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, give 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 189.

`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/(sash.cpu) 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 189, or try again later.

Other Errors Loading the Miniroot

```
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)sash.cpu: file not found
```

or

```
open(bootp()server:CDdir/dist/sa(sash.cpu)) failed, errno=2
Unable to load bootp()server:CDdir/dist/sa(sash.cpu): 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.

```
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 message only occurs with the Indy workstation. If you are not using an Indy workstation but see a similar error, refer to the discussion of the next error. This error (or the next one) occurs when you try to load the miniroot after a power failure or system restart has occurred during an installation. If you had attempted a system restart instead of loading the miniroot, you would have automatically been placed in the version of Inst that is in the previously-installed miniroot.

Take one of the following corrective actions:

1. Enter **c** if you want to install software with the currently loaded miniroot.
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 want to use the *fx* command to correct boot information and boot from the root partition, or if you want to abort the installation and restart the system.

Note: Entering **c** and quitting Inst fixes the boot information. You can then restart the system after Inst is loaded without using the *fx* command. If you are familiar with *fx* and want to use it, refer to the procedure documented in “Using fx to Restore the Swap Partition” on page 200.

```
root and swap are on the same partition. This is most likely
because a previous installation was in progress. If so, you may
continue the boot into the miniroot. Otherwise the partition
info needs to be corrected. Do you wish to continue booting (y
or n)
```

This is the version of the previous error message for systems previous to the Indy. If you see this error after you have given the command to copy the miniroot to the swap partition, it may be that the power failed or the system was reset during installation, and the miniroot is still in the swap partition.

First decide whether you need to return to Inst to complete your installation, or are ready to restart your system. If you need to return to Inst, answer *y* to the question. The Inst Main Menu should appear, and you can finish your installation. To restart your system and cause it to boot normally from the root partition, enter *n* and you will be returned to the PROM Monitor where you can choose to start the system.

Errors While Starting an Installation Session

The error discussions in this section are grouped as follows:

- “Wrong Diskless Modes” on page 172
- “Errors Starting Live Installation” on page 173
- “Inst Library libinst.so Errors” on page 173
- “Errors in the Distribution” on page 174

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: /root appears to be a diskless
client tree, since the file /root/var/inst/.client is present.
If you are certain that /root is not a diskless client tree,
remove the file /var/inst/.client and restart inst, otherwise
restart in client mode using client_inst(1m).

ERROR : Unable to start inst: / appears to be a share tree for
diskless client since the file /var/inst/.share is present. If
you are certain that / is not a diskless share tree, remove the
file /var/inst/.share and restart inst, otherwise restart in
share mode using share_inst(1m).
```

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*(1M) and *client_inst*(1M).

If the target has been previously created as a diskless tree, then continuing with a normal (non-diskless) installation would severely corrupt the installed software. You should only attempt diskless installations using *share_inst*(1M) and *client_inst*(1M).

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 Installation

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 72.

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 *eoe1.sw.unix* 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 the *libinst.so* file is missing.

```
852:swmgr: rld: Error: unresolvable symbol in swmgr:
post__15VkDialogManagerFPCcPFP10_WidgetRecPvT2_vN22PvT1P14VkS
impleWindow
```

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 host name 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 189 for information on resolving this problem.

`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 non-existent 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 of 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 199 for more information.

Pre-Installation Check Errors

When you give the *go* command, Inst executes the pre-installation 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 (in these examples on the / and */usr* filesystems), or select fewer subsystems for installation.

Note: 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 a live installation and, in some cases, until you reboot the computer.

If you are running a live installation, you may encounter a situation where there is enough available disk space for all the new software, but not enough additional temporary disk space to accomplish the installation. In this situation, try closing some applications, and then giving the "go" command again. If there is still not enough space, you may have to run the installation in the miniroot.

```
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*.

```
directory /pathname is write-protected
```

```
nfs-mounted directory /pathname is write-protected
```

```
filesystem /pathname is mounted read-only
```

```
nfs-mounted filesystem /pathname is read-only
```

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 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 an online books or manual pages. Use the *keep* command to de-select those products.

Errors While Installing and Removing Software

This section contains the following subsections:

- “Disk Space Errors” on page 179
- “Sub-Command and Exitop Errors” on page 181
- “Network Timeout Errors” on page 182
- “Archive Corrupt Errors” on page 182
- “Device Busy Errors” on page 183

These errors cause the following Error/Interrupt menu to appear automatically:

```
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>
```

If the pre-installation 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 doesn’t work, or you are unable to correct the problem, you can choose *stop* to cancel the installation immediately and return to the main menu.

Caution: If you *stop* the installation, the current image in progress (such as *eoe1.sw*) will be in an inconsistent state (partially installed/removed). The installation history will not have been updated for these subsystems (*eoe1.sw.**). You are strongly advised to either re-install these products (just select *Go* at the main menu to re-start 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:

```
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. For example:

```
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 using IRIX 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 may want to truncate or remove it. Since it is in use, you must notify the daemon:

  ```
  # 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 file systems with these commands if you are using 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 path name 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, 2000 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 file systems 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.

Sub-Command and Exitop Errors

As part of the installation procedure, Inst executes *sub-commands*. These are UNIX shell commands that perform special initialization functions specific to each product. For example, some products use sub-commands to install a custom icon in the system Icon Catalog. Some sub-commands, 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 sub-command fails during the installation of a specific product, an interrupt menu is also presented. The sub-commands 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 won't 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 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 Timeout 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 189 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 189).

If you are performing a live installation, you may need to use a newer version of the installation tools, since older versions of Inst cannot always read more recent software distributions (but not vice-versa) if the distribution format is upgraded. 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 re-invoke 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 On 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 due to 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 and Other Errors" on page 63 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 185
- "Overriding Conflicts" on page 189

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:

- "Pre-Existing Conflicts" on page 186
- "Required Product Conflict" on page 186
- "Prerequisite Conflict" on page 187
- "Back Prerequisite Conflict" on page 187
- "Incompatible Products Conflict" on page 188
- "Downgrade Conflict" on page 188

Pre-Existing Conflicts

Unresolved product dependencies and incompatible installed products may already exists on the target before the current Inst session. These conflicts are known as pre-existing 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 eoe1.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 and Other Errors” on page 63.
2. Choose to ignore pre-existing 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 189.

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 (xxxxxxxxxx)
```

This type of conflict occurs when the distribution contains a required subsystem that is not installed and not marked for install. 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 (xxxxxxxxxx)
```

This type of conflict occurs if you mark a required subsystem for removal. In this case, the only solution is to not remove 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 - xxxxxxxxxx)
```

To resolve the above conflict, you have two options. The first is to not install the product whose requirements are not met. 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 51 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 choose to remove all of the products that depend on it also.

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. You must not install 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, the user can override the conflicts and continue with the installation by setting the *rulesoverride* preference. Overriding conflicts leaves a pre-existing conflict on the target, which can be seen if the *show_existing_conflicts* preference is set to *true*.

This procedure is recommended only for extreme cases and for knowledgable users, since it can introduce unfulfilled product dependencies or install incompatible products onto the system.

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 - xxxxxxxxxx) (not on current
distribution)

Inst> set rulesoverride true
Inst> go
```

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
   ```

   where *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 isn't returned, you've specified the wrong *server*, there is a network problem, or *user* isn't a valid user ID. (See"Configuring an Installation Account" on page 19 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*(1C). A simple test of this connection can be done by using *ping*(1M):

   ```
   % /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
   %
   ```

   where *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 using 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 non-volatile random access memory (NVRAM) doesn't 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 IP address in the NVRAM with this command:

```
>> printenv netaddr
```

If the four-part number returned from either command doesn't 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*(1M) access (required for miniroot installation only).

   "Enabling TFTP Access on an Installation Server" on page 18 describes the procedure for verifying that the installation server has been modified to allow *tftp* access.

   To get more debugging information, add the *–l* argument to the *tftp* line in */etc/inetd.conf* and restart *inetd*(1M). 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*(1M) packets.

   "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*(1M). 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 Advanced Site and Server Administration Guide* and the *NFS and NIS Administration Guide and Man Pages*.

Network Problem Diagnosis During an Inst Session

If the network is very slow (usually indicating network problems), Inst may appear to be frozen for long periods (much greater than the timeout 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 timeouts or retries, are 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’ll 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. There’s probably a serious network problem that needs to be resolved before we can continue. Either resolve the network problem and continue the install, or cancel the install and try again later.
>
> On a very slow network, changing the network timeout (by setting the preference “timeout” to the new timeout 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 189" 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 host's name was mistyped, that there's 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 reason. 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 there’s something wrong with the file */etc/services* or the services map on the NIS server.

Network Problem Diagnosis During Miniroot Installation

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
```

where pathname is the distribution directory; for example, *sa* may be in *dist:/irix/5.3.*

Chances are the file does exist and this is probably a network problem—trying to *bootp* through too many gateways. Since 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 multi-homed 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 your *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 “Preliminary PROM Monitor Procedures” on page 32 for information on the *netaddr* variable).

```
-- Error 7 while loading
network(0)bootp()mach1:/path/sa(sashARCS)
```

```
-- UX:csh: ERROR: ./.swindow - Command not found
```

[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 IP12 prom and 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're booting from the command monitor, be sure to specify the *sa* file and not just the distribution directory containing the *sa* file.

`-- File sash.`*`XXX`*` not found in `*machine*`:/`*path*`/sa, directory contains (listing)`

You specified an unknown or unsupported machine type *XXX* when selecting the *sash* from which to boot. See "Understanding Miniroot Installations" on page 30 for more information.

`-- No remote connection`

This is probably an internal error resulting from trying to access a remote connection which 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's 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:

1. If IRIX is running, give 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.
2. If you are in the miniroot, escape to a shell with the *shroot* command and give 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.
3. If you are in the Command Monitor, give 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 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*(1M) 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 is done by the command *cdinstmgr*(1). To verify that the CD is mounted, use the *df*(1) command below. If you are using a local CD-ROM, escape to a shell. If you are using a remote CD-ROM, give 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 local CD-ROM, you should see */CDROM*. For 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. If the CD is not mounted and you are using an installation server, verify that *cdinstmgr*(1) is running.

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 doesn't 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 *eoe1* 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 *distcp*(1M)] 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 *distcp*(1M)].

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*(1) to read the release notes.

Using fx to Restore the Swap Partition

The following procedure is an example of a use of the *fx* command. The procedure shown corrects the condition described in “Other Errors Loading the Miniroot” on page 169, in which you receive the error message “root and swap are on the same partition.” In this procedure, the boot information is modified to cause you to boot from partition 0 (the root partition) instead of partition 1 (the swap partition) where the miniroot currently resides. (This discussion assumes that the 4D1-4.0 or later version of *fx* is used.)

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 reference page for *fx*(1) for complete information on using *fx*.

1. Enter the command monitor (see “Preliminary PROM Monitor Procedures” on page 32 for details on how to enter the command monitor on your system).

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 the appropriate command.

     If your CPU type (*cpu*) is IP4, IP5, or IP6, the command is:

     ```
     >> boot -f dksc(cntlr,unit,8)sash.cpu
     dksc(cntlr,unit,7)stand/fx.cpu --x
     ```

     If your CPU type is IP7 or IP9, the command is:

     ```
     >> boot -f dksc(cntlr,unit,8)sash.cpu
     dksc(cntlr,unit,7)stand/fx.IP5 --x
     ```

     If your CPU type is IP12 or IP17, the command is:

     ```
     >> boot -f dksc(cntlr,unit,8)sashcpu
     dksc(cntlr,unit,7)stand/fx.cpu --x
     ```

     If your CPU type is IP19, IP20, or IP22, the command is:

     ```
     >> boot -f dksc(cntlr,unit,8)sashARCS
     dksc(cntlr,unit,7)stand/fx.ARCS --x
     ```

     where *cntlr* and *unit* are the controller and unit numbers of the CD-ROM drive (see "Getting CD-ROM Device Numbers" on page 24).

   - To invoke *fx* from a CD with installation tools that is mounted on an installation server named *server*, enter the appropriate command.

     If your CPU is IP4, IP5, IP6, IP12, or IP17, the command is:

     ```
     >> boot -f bootp()server:CDdir/stand/fx.cpu --x
     ```

     If your CPU type is IP7 or IP9, the command is:

     ```
     >> boot -f bootp()server:CDdir/stand/fx.IP5 --x
     ```

     If your CPU type is IP19, IP20, or IP22, the command is:

     ```
     >> boot -f bootp()server:CDdir/stand/fx.ARCS --x
     ```

     where *CDdir* is the mount point directory for the CD (for example, */CDROM*) and *cpu* is the CPU type of your workstation.

3. After *fx* is invoked, messages, prompts, and menus appear. A sample session is shown below. In this sample session, the default system disk device name, controller number and drive number chosen by *fx* (dksc, 0, and 1 respectively) are used. Unless your workstation is configured in a non-standard way, the defaults chosen by *fx* will be 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 IP6, Aug 23, 1991
fx "device-name" = (dksc) <Enter>
fx: ctlr# = (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/    [exe]rcise/   [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*(1M))
- 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

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 007-1364-050.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 415-965-0964
- 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