Chapter 4

Using the Main Menu

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

This chapter contains these sections:

- “The Software Installation Procedure”
- “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.

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 enter commands from other menus during installation sessions.

Note: If you are an experienced Inst user, you might prefer to use the "Condensed Installation Instructions" on page 137 instead of this procedure.

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

1. Start Inst.
2. Specify the distribution source 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 4-1 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 4-1, the trigger is the *from* command.

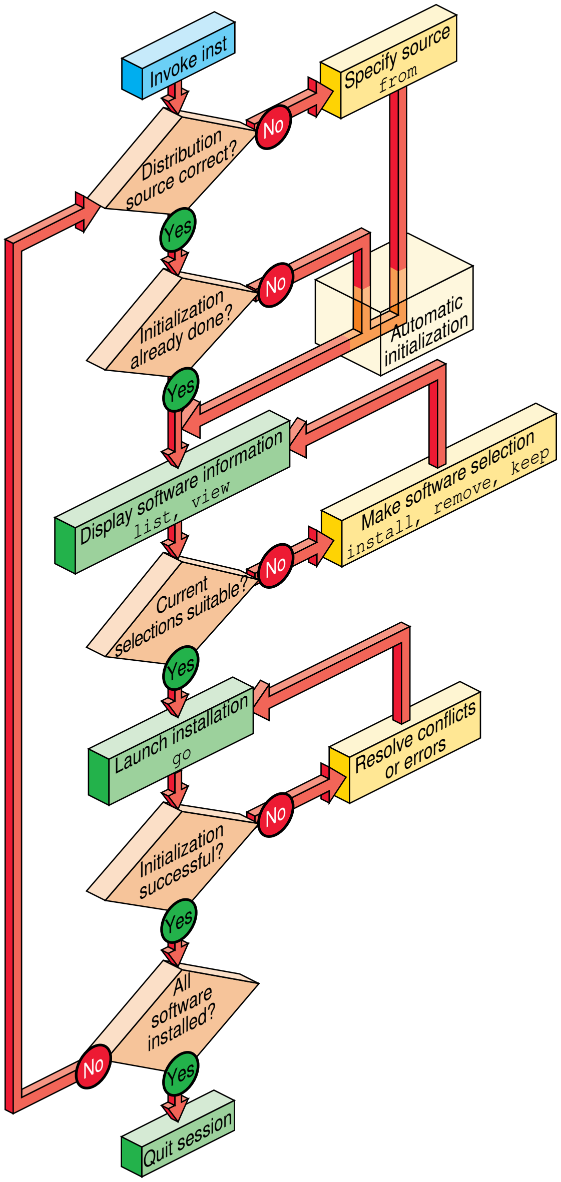
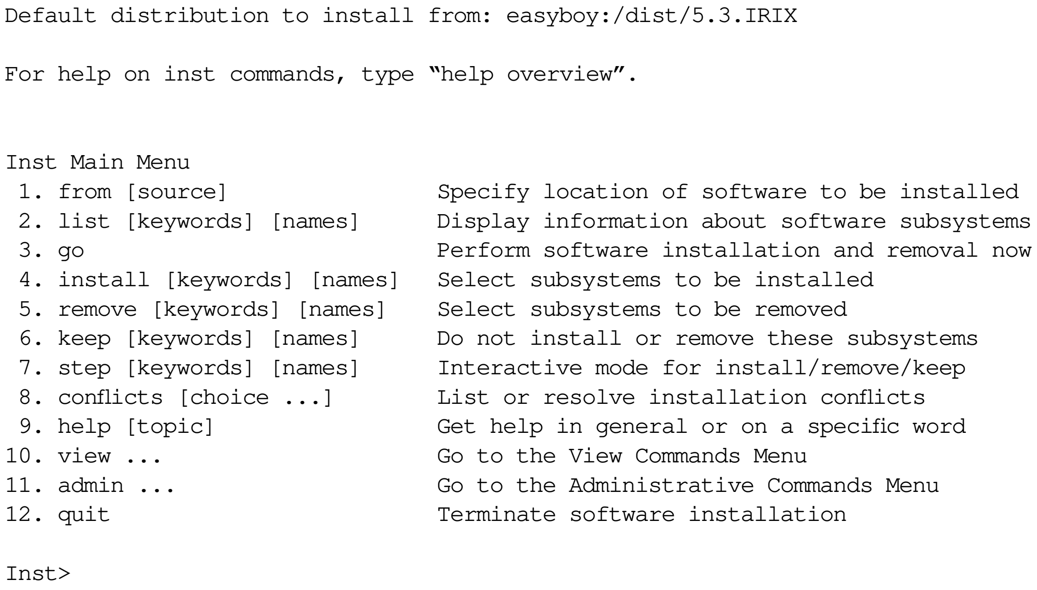


Figure 4-1 Summary of the Installation Procedure

Step 1: Invoking Inst

For miniroot installations, Inst is automatically started when the miniroot is loaded (see “Starting a Miniroot Installation” on page 30). For live installations, Inst is started from the command line of an IRIX shell (see “Starting a Live Installation” on page 37). After Inst is started, it displays the default distribution source and the Main menu, as shown in Figure 4-2.



```
Default distribution to install from: 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 4-2 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 started. 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 enter 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 83):

from

Enter the *from* command with no arguments to display a selection of former distributions from which to choose; type the number of a selection to specify it as the distribution. If the distribution that you want 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_6.2/dist
```

This form of *from* 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_6.2/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 enter 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 software on the target using the *view* command (see "Using the View Commands Menu" on page 66 for additional ways of using *view* to control *list* displays). As you select software for installation or removal (step 4 of this procedure), *list* displays are automatically updated to reflect your selections.

Using the list Command

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

Example 4-1 illustrates a *list* entry and the legend that precedes *list* displays. In this example, the distribution contains a software patch.

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

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

Example 4-1 The list Legend

As Example 4-1 shows, the *list* legend provides information about software components and about actions taken during the installation session:

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

Example 4-2 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 6.2 Cache File System
```

Example 4-2 Sample list Display of Distribution Software

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

Displaying Software Installed on the Target

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

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

Example 4-3 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 4-3 Sample list Display 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 4-1). Columns appear empty when a tag is unassigned or when the tag is masked by the current view setting (see "The target and distribution Commands" on page 67 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 4-1).

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 4-2 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 4-3 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 4-2, *onc3_eoe*, does not appear in the target list in Example 4-3. 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 4-4 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 4-4 Sample list Display 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 4-5. The summary identifies the target filesystems to receive the selected software, estimates the additional space (or space savings) that will result if the selected software is installed, calculates the space overhead required by installation processing, and reports the amount of space that will be available on the target filesystems after the installation. Estimates in disk space summaries are based on the current installation selections.

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

Final projected free space                563156
```

Example 4-5 Sample Disk Space Summary

Step 4: Making Software Selections

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

In most cases, the selection step is needed to adjust software for the disk capacity of the target and the needs of its users. However, in some cases it is possible to accept the automatic selections that Inst made during initialization. These selections are described in “Accepting Automatic Selections” on page 52.

Note: If the automatic selections described on page 52 are suitable for your circumstances, you can accept these selections and go directly to step 5 of this procedure.

Using the install, remove, and keep Commands

The *install*, *remove*, and *keep* commands select software products and their components for a particular action. No action is taken until you enter the *go* command (see “Step 5: Launching the Installation” on page 53), 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 90 for details.

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

- Usually, Inst has already made some installation selections automatically (see “Accepting Automatic Selections” on page 52). 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 4-3). 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 44).
- It is sometimes useful to alternate the selection process with the launch process (see “Step 5: Launching the Installation” on page 53), 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 the *install*, *remove*, or *keep* commands, the action is taken on all subsystems of the product or image.

These sample entries explain the effects of selection command entries:

```
Inst> install sysmon
```

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

```
Inst> 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 4-6. Your list of arguments can extend to a new line, as long as you do not end the line with a blank space.

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

Example 4-6 Multiple Arguments to the install, remove, and keep Commands

Using Arguments in Selection Commands

You can use the names of products and their components, metacharacters, and keywords as arguments to the *install*, *remove*, and *keep* commands to save time (see "Using Command Shortcuts" on page 85 for more information). These sample entries illustrate the use of 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 eoe*
```

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 enter a *list* command during the selection process whenever the `Inst>` prompt is displayed. The first column of the listing and the disk space summary information 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 44). If you accept the automatic selections, you can omit step 4 and go on to “Step 5: Launching the Installation” on page 53.

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

```
Inst> list i
```

Example 4-7 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 eoe.sw.gfx_lib bdr      19+  Graphics Execution Libraries
i  U eoe.sw.irix_lib bdr    991+  IRIX Execution Libraries
i  U eoe.sw.unix bdr       6674+  IRIX Execution Environment
i  U eoe.sw.cdrom d         241+  CD-ROM Support
i  U eoe.sw.crypt d          25+  Security Administration Utilities
i  U eoe.sw.gltools d        80+  Graphics Library Tools

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

Final projected free space           499639
```

Example 4-7 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 54 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 4-8 Successful Installation Messages

Notice that Example 4-8 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 56. 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 B, “Troubleshooting Installations,” for additional error-handling information).

Using the conflicts Command

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

Example 4-9 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

DLPI Execution Environment cannot be removed because other products
depend on it:
  2a. Do not remove DLPI Execution Environment
  2b. Also remove 4DDN Software
```

Example 4-9 Sample Conflicts Message

To resolve a conflict, enter the *conflicts* command and your choice of resolutions as command arguments, as shown in Example 4-10.

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

Example 4-10 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 4-8), you can continue with the session or go on to "Step 7: Quitting the Installation Session" on page 56.

Correcting Other Installation Errors

If Inst is unable to execute *go* for reasons other than conflicts, it posts error messages that suggest the cause of the error and presents choices for your next action. Errors can occur during pre-installation checking, while software is being installed and removed, or during exit operations.

Example 4-11 illustrates an error that occurs in pre-installation checking.

```
Inst> go

ERROR: Subsystems cannot be installed/removed because they contain
files in write-protected directories.

ERROR: Subsystems cannot be installed/removed because they contain
files on read-only NFS-mounted filesystems.

Inst>
```

Example 4-11 Sample Error in Pre-Installation Processing

Most error messages contain information that suggests the needed corrective action; however, if you need additional information to correct an error, see Appendix B, "Troubleshooting Installations."

Example 4-12 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 /swmgr/products/prod1/images/prod1.sw is corrupt

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

Interrupt>
```

Example 4-12 Sample Error in Installation Processing

Notice from Example 4-12 that if you encounter errors during installation, Inst displays the Interrupt menu (explained in “Using the Interrupt Menu” on page 80). 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 B, “Troubleshooting Installations.”

Example 4-13 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 4-13 Sample Error in Exit Operations

See Appendix B, “Troubleshooting Installations” 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 this 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 enter 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 enter 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 4-14 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 54 for information on conflict resolution.

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

If Rebooting Is Needed

If you install products that require a system reboot to put the software into effect, you see the message in Example 4-15.

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

Inst>
```

Example 4-15 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. 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 see informational messages similar to this:

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

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. Ignore the previous installation session.

2. Restore the previous installation session and let me look at it.

3. Retry the previous installation session.

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

Figure 4-3 The Checkpoint Restart Menu

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

- 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 started in a new session.

If a Checkpoint Restart Fails

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

- A missing software distribution

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

  This failure occurs if the distribution source that was saved in the checkpoint restart file no longer 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 54 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.

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 25). 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 25):

   ```
   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, enter 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/rpc* file, for example). When the target system contains modified configuration files, Inst preserves the modified files during the installation in one of two ways:

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

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

  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 4-16, 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 4-16 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:

   ```
   # versions changed
   ```

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

   ```
   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
```

Example 4-17 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.